

O TK 90X é a última palavra em microcomputador!

Além da vantagem do custo, apresenta simplicidade de uso capaz de habilitar tanto os jovens como os adultos a desfrutar de todos os seus múltiplos recursos.

É fantástica a sua capacidade de manipular os diversos tipos de programas educacionais.

Neste livro são apresentados 45 programas, dos quais 30 totalmente comentados. À medida que eles aparecem, existe uma metódica e seqüencial explicação de todas as instruções da linguagem BASIC que é entendida pelo TK 90X.

Os princípios envolvidos e as técnicas especiais utilizadas são totalmente descritos, permitindo ao iniciante compreender rapidamente como funcionam os programas enviados ao TK 90X.

Este livro permite que você compreenda, no futuro, qualquer programa escrito na linguagem BASIC no TK 90X.

ISBN 85-213-0321-1

**BASIC NO TK 90X**

nobel

# BASIC



## NO

# TK 90X



Victor Mirshawka



nobel

**BASIC NO TK 90X**

M659b Mirshawka, Vitor, 1941-  
BASIC no TK 90X / Victor Mirshawka. — São  
Paulo : Nobel, 1935.

Bibliografia.  
ISBN 85-213-0321-1

1. BASIC (Linguagem de programação para compu-  
tadores) 2. Microcomputadores - Programação 3.  
Programas de computador 4. TK 90X (Computador) -  
Programação I. Título.

85-1088

17. CDD-651.8  
18. -001.6424  
18. -001.642

Índices para catálogo sistemático:

1. BASIC : Linguagem de programação : Computadores : Pro-  
cessamento de dados 651.8 (17.) 001.6424 (18.)
2. Microcomputadores : Linguagem de programação : Proces-  
samento de dados 651.8 (17.) 001.642 (18.)
3. Microcomputadores : Programação : Processamento de da-  
dos 651.8 (17.) 001.642 (18.)
4. Programação : Microcomputadores : Processamento de da-  
dos 651.8 (17.) 001.642 (18.)
5. TK 90X : Computadores : Programação : Processamento de  
dados 651.8 (17.) 001.642 (18.)

## VICTOR MIRSHAWKA

Professor titular de Cálculo Numérico, Estatística e Pesquisa Operacional  
da Faculdade de Engenharia da Fundação Armando Álvares Penteado.  
Professor titular de Cálculo Numérico e Ciência da Computação, Estatística  
e Pesquisa Operacional da Universidade Mackenzie.

Professor associado do departamento fundamental da Escola de  
Engenharia Mauá.

Mestre em Estatística pela Universidade de São Paulo.

# BASIC NO TK 90X

1ª edição  
1ª reimpressão  
1986

# APRESENTAÇÃO

© 1985 Livraria Nobel S/A

Livraria Nobel S.A.  
Rua da Balsa, 559  
02910 – São Paulo – SP

## É PROIBIDA A REPRODUÇÃO

Nenhuma parte desta obra poderá ser reproduzida sem a permissão por escrito dos editores através de qualquer meio: Xerox, fotocópia, fotográfico, fotomecânico. Tampouco poderá ser copiada ou transcrita, nem mesmo transmitida através de meios eletrônicos ou gravações. Os infratores serão punidos através da Lei 5.988, de 14 de dezembro de 1973, artigos 122-130.

Impresso no Brasil / Printed in Brazil

Entre os objetivos que tive em vista ao escrever este livro quero destacar o mais importante ou seja aquele de apresentar um texto que permita atender as necessidades de todas as Escolas na implantação dos seus cursos iniciais de Informática estruturados sobre o TK 90X.

Assim a Microdigital fabrica o fantástico TK 90X, a Livraria e Editora Nobel S.A. possibilita a produção e a distribuição do livro e a mim cabe a missão mais simples ou seja elaborar o próprio.

Espero que este livro, em conjunção com o manual que acompanha o TK 90X, permita a você estimado(a) leitor(leitora) dominar a linguagem BASIC toda especial do seu TK 90X.

Aliás já escrevi vários livros para o TK 85 e para o TK 2000 II (você já os conhece?) e este vai lhe dar condição de usar o único micro pessoal lançado no Brasil até agora, que realmente merece a classificação de "software machine", um título incomum porém merecido, devido a facilidade do seu uso, os seus grandes recursos e por que não destacar o seu preço acessível.

Recentemente tive a felicidade de conhecer a figura eminente do padre e professor Waldemar Valle Martins que me presenteou com um dos seus livros - "A liberdade de ensino" - que contém importantes reflexões a partir da situação do Brasil.

# AGRADECIMENTOS

Uma das primeiras frases do livro é: "Entende-se por liberdade de ensino a prerrogativa de escolher um determinado gênero de educação".

Sem entrar na análise do conteúdo complexo desta assertiva, o que posso dizer ilustre educador Waldemar Valle Martins é que além das Escolas, das revistas, dos jornais, do rádio, da televisão e principalmente dos templos que possibilitam a educação surge agora um outro fantástico meio para chegar a mesma ou seja o microcomputador TK 90X.

Este livro é um guia para dominar o manejo do seu TK 90X...

Um micro como o TK 90X realmente não discrimina ninguém.

Todos podem adquiri-lo e utilizar no "refúgio" domiciliar ou na pior das hipóteses utilizá-lo nas escolas do 1º, 2º e 3º graus ou ainda nas empresas.

O TK 90X permite obter resultados que varrem não apenas a área educacional, mas também a técnica, o lazer e os pequenos negócios.

O livro "BASIC NO TK 90X" é o primeiro de uma grande série que pretendo lançar para possibilitar a todos o completo domínio desta máquina fabulosa - TK 90X.

ANTES DE COMEÇAR O LIVRO NÃO ESQUEÇA  
QUE O SEU TK 90X É TOTALMENTE COMPATIL  
COM O ZX-SPECTRUM



Foi contínua e cada vez mais indispensável a ajuda da minha querida esposa e amiga Nilza Maria e dos meus três filhos Victor Junior, Sergio e Alexandre (Sacha para os íntimos, que agora já ajudou...).

Eu sei que para aqueles que me prestigiam usando os meus livros de Informática, a primeira parte da frase acima parece um "repeteco" um tanto sonso...

Pois então vou melhorar o meu depoimento dizendo que Nilza Maria já me ajudava com os meus livros na minha primavera (escrevi o meu primeiro livro em 1958) e no verão da minha existência colabora revendo, corrigindo e indicando a melhor forma de apresentar um assunto.

Estou a um passo de entrar no outono da minha vida e com que alegria vejo o primeiro livro publicado pelos meus filhos Victor Junior e Sérgio.

Estamos certos Nilza Maria e eu, que se Deus nos permitir chegar ao inverno da existência ele não será frio, mas muito ao contrário acalenta-dor e repousante por termos deixado um bom exemplo, pelo menos para os nossos filhos.

Muito obrigado ao amigo George Kovari, diretor da Microdigital, por mais uma vez ter me engajado neste projeto muito mais educacional do que comercial.

# SUMÁRIO

Estou seguro que o TK 90X irá modificar de forma substancial o ensino, dinamizando-o e principalmente colocando a disposição dos que estudam e trabalham os mais fantásticos recursos para texto, cálculos, gráficos e som.

"Arigatô", Toshiro Iqueda, datilógrafo, editor e amigo de muitos anos (datilografou todos os meus livros a partir do terceiro e hoje já tenho mais de cinquenta títulos publicados...).

Finalmente e com muita sinceridade sou grato mais uma vez aos amigos da Livraria Nobel S.A., Ary K. Benclowicz e Luigi Zamboni por acreditarem em tudo o que escrevo e por investirem nisto de corpo e alma colocando toda a empresa a minha disposição.

Assim até parece fácil escrever e editar um livro...

Capítulo 1	
Palavras-chave ou seja ensinando ao TK 90X o que deve ser feito .....	1
Capítulo 2	
Como se pode imprimir (PRINT) números na tela do seu aparelho de T.V. ....	5
Capítulo 3	
Imprimindo palavras na tela .....	9
Capítulo 4	
Fazendo operações matemáticas no TK 90X ....	13
Capítulo 5	
Obrigado TK 90X pela sua memória!!!! .....	19
Capítulo 6	
Lembrando das letras e das cadeias de letras	27
Capítulo 7	
Finalmente escrevendo o seu primeiro programa!!!! .....	31
Capítulo 8	
Aprendendo a editar e corrigir erros .....	37
Capítulo 9	
O comando INPUT, um novo e futuro grande amigo seu .....	
Capítulo 10	
Divertindo-se com a programação .....	47
Capítulo 11	
Guardando e carregando os seus programas....	55

Capítulo 12	
Como ter uma possibilidade ligada a um acontecimento aleatório .....	63
Capítulo 13	
Fazendo decisões com o auxílio do TK 90X ..	75
Capítulo 14	
Utilizando subrotinas para inclusive obter economia de digitação .....	87
Capítulo 15	
Movendo coisas na tela .....	93
Capítulo 16	
Conectando e cortando STRINGS .....	103
Capítulo 17	
Preparando-se agora para (FOR) o seu próximo (NEXT) grande truque no TK 90X .....	
Capítulo 18	
Movendo-se na tela com o auxílio do FOR-TO/ NEXT .....	117
Capítulo 19	
Usando caracteres gráficos que permitam simular uma explosão .....	129
Capítulo 20	
Desenhando com linhas bem finas .....	137
Capítulo 21	
Incrementando ao máximo a alta resolução ..	157
Capítulo 22	
Matrizes .....	199
Capítulo 23	
Programas adicionais .....	205
Índice remissivo .....	231

## Capítulo 1

### Palavras-chave ou seja ensinando ao TK 90X o que deve ser feito

Um micro não é nem um pouco tão inteligente como você...

Ele precisa receber instruções claras de você sobre tudo o que deve fazer.

Você sabia que os micros não são tão inteligentes como as pessoas?

O TK 90X não pode fazer nada por si próprio.

Ele necessita de você caro(a) leitor (leitora) para dizer-lhe o que deve ser feito.

Os humanos, para entender-se com os micros e principalmente para dar-lhes ordens, desenvolveram uma série de linguagens, *chamadas de alto nível*, tais como o FORTH, PASCAL, COBOL ou o BASIC.

Todas tem suas próprias regras de sintaxe, as quais são muito mais simples e restritas do que as que exige um idioma falado ou escrito entre pessoas como é o caso do Português.

Cada uma das palavras que integram estas linguagens de alto nível não podem ser entendidas diretamente pelo microprocessador (ou seja o "cérebro") do TK 90X.

Um microprocessador só pode executar micro instruções do seu particular idioma, cujo nome é *Linguagem de máquina*.

Desta forma qualquer palavra de um idioma de *alto nível* está formada por várias micro instruções concatenadas de tal forma que, finalmente, realizam o que o(a) programador(programadora) espera das mesmas.

Todos os idiomas de alto nível, citados acima, foram desenvolvidos por pessoas de língua inglesa e conseqüentemente as palavras-chave são em Inglês.

A linguagem de alto nível que iremos aprender para dar ordens ao TK 90X chama-se BASIC e no lugar de ter milhares de palavras (algumas com duplo e até triplo sentido) como o Português, tem aproximadamente 90 palavras.

É de alto nível e ao mesmo tempo muito simples...

Assim como você diz ao seu "cãozinho"-fila "senta" ou "deita aí!" você (com um pouco mais de delicadeza...) pode pedir para o TK 90X que entre em "ação" usando palavras simples tais como:

PRINT, LET, LIST etc.

Naturalmente, ainda não dá para mandar estas ordens por via oral.

Você ordenará isto para o seu TK 90X utilizando o seu teclado.

Acho entretanto, que em alguns anos estas instruções poderão ser enviados através do movimento sincronizado da sua língua dentro da sua boca acompanhada de inspirações e expirações...

As palavras mais importantes que o TK 90X entende são as chamadas *palavras-chave*.

Todas as palavras-chave do TK 90X estão impressas na cor branca nas teclas, perto da letra principal.

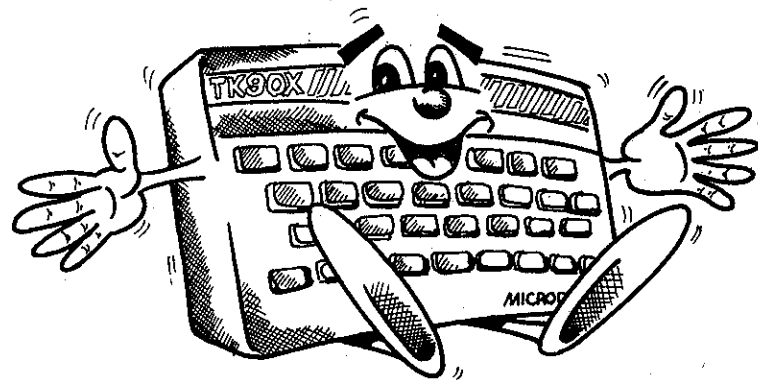
Assim por exemplo você encontra a palavra-chave LET na tecla L a palavra-chave PRINT na tecla P, a palavra-chave LIST na tecla K, etc.

Todas as palavras chave em branco no seu TK 90X são chamadas de *comandos*.

São elas que indicam ao TK 90X o que deve ser feito como PRINT (imprima), LOAD (carregue), NEW (novo), RUN (execute), GOTO (va para), etc.

Um TK 90X pode "carregar", "executar" e "ir para"?

Não se amofine respeitado(a) leitor(leitora) vamos explicar tudo nos mínimos detalhes, daqui para frente...

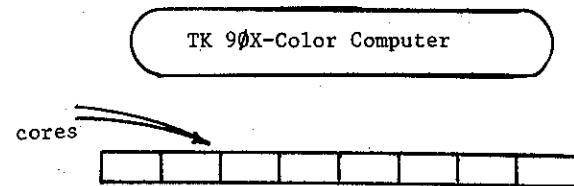




## Capítulo 2

### Como se pode imprimir (PRINT) números na tela do seu aparelho de T.V.

Esteja seguro que a tela da T.V. esteja limpa e contenha apenas a mensagem



Olhe agora com atenção no teclado e veja que no lado direito do teclado você tem a tecla **ENTER**.

Aperte a mesma.

Note que a mensagem

TK 90X-Color Computer

desaparece e no seu lugar surge um pequeno retângulo (cursor) piscando **K**.

Caso você não possa afirmar que está vendo um **K** então precisa ajustar o seu aparelho de T.V.

O **K** piscando é uma forma do TK 90X lhe dizer que ele quer que você lhe envie uma *palavra-chave*.

Para não decepcionar o seu TK 90X aperte por exemplo a tecla P.

Aparecerá na tela a linha

PRINT

Deu para você perceber que o  "piscante" foi substituído agora por um  piscando?

Uootimo!

O  piscando significa que o TK 90X está esperando agora que você tecla uma letra ou um número.

Vamos entrar com o número 10.

Para tanto você encontrará o 1 na primeira tecla a esquerda e acima no teclado e o 0 (zero) na outra extremidade e a direita.

A barra cortando o zero (0) é para que você não faça confusão com a letra "o".

Bem aperte a tecla "1" e em seguida a tecla "0" e a seguir aperte a tecla ENTER.

O que aconteceu?

Resp.: Você disse para o TK 90X que imprimisse

(PRINT) o número 10 na tela da T.V.

E é o que realmente vai acontecer...

Tente agora imprimir outros números na tela e para tanto tecla o seguinte:

<input type="text" value="PRINT"/>	seguido por	<input type="text" value="33447"/>	e por	<input type="text" value="ENTER"/>
<input type="text" value="PRINT"/>	seguido por	<input type="text" value="2237"/>	e por	<input type="text" value="ENTER"/>
<input type="text" value="PRINT"/>	seguido por	<input type="text" value="999"/>	e por	<input type="text" value="ENTER"/>

Você perceberá que cada número novo é impresso em uma nova linha abaixo do último número.

Não se preocupe (por enquanto) se algo ocorrer errado.

Neste caso tome a decisão mais elementar: desligue o TK 90X da tomada e ligue novamente.

Podemos começar tudo de novo e teclar certo daí para frente para não ter que repetir esta estranha operação de inicialização...

Porém, se você entrar com um número bem grande como

987654321

o seu TK 90X irá apresentá-lo de outra forma.

O TK 90X usa uma notação especial para os números grandes e assim aparecerá na tela.

9.8765432 E+8

O E+8 significa: mova o ponto decimal "." (é a nossa clássica vírgula decimal ",") oito posições para a direita quando então volta-se a situação original 987654321 (só que o 1 não aparece na tela).

Assim se tivéssemos o número 9.87654321 E-5 isto significa que temos um número muito pequeno e deve-se mover o ponto decimal *cinco posições para a esquerda* ou seja o número é 0.0000987654321.

Geralmente você não irá precisar desta notação, a não ser que esteja elaborando programas onde são feitos cálculos que resultem em números muito grandes ou muito pequenos.

Costuma-se chamar esta notação de *exponencial*.

Não esqueça portanto que:

→ Diz ao TK 90X que você terminou de teclar algo e deseja que o mesmo faça algo.

→ Diz ao TK 90X para imprimir algo na tela. Por enquanto só imprimimos números, mas logo em seguida você irá aprender a imprimir palavras, frases e respostas para operações.

Ø Executado Ø:1 Ø TK 90X imprime esta mensagem na tela, para lhe indicar que já fez o que você lhe pediu ou ao menos ele pensa isto...

## Capítulo 3

### Imprimindo palavras na tela

Cursor **K** → O TK 90X está esperando que você aperte uma palavra-chave.

Cursor **L** → O TK 90X irá interpretar a próxima coisa que você irá teclar como uma simples letra ou número.

Cada vez que se aperta uma tecla, algo sucede (ou ao menos predispõe-se a suceder) na tela.

Neste sentido, o seu melhor colaborador é o *cursor* que está representado na tela por uma letra que está piscando e que lhe indica o *modo* no qual o TK 90X está predisposto a trabalhar.

Você já conhece os modos **K** (de keyword = palavra-chave) e **L** (de letter = letra) e irá aprender adiante o uso dos modos **G** (de graphics = gráfico), **C** (de capitals = maiúsculas) e **E** (de extended = ampliado ou estendido).



Chegou o momento de fazer algo mais interessante do que simplesmente ver números na tela.

Aperte a tecla "P" para obter a palavra-chave PRINT na tela.

Agora procure a tecla marcada com as palavras em vermelho

SYMBOL  
SHIFT

Você encontrará esta tecla a direita na linha de baixo do teclado.

Mantenha esta tecla pressionada **para baixo** com um dedo e aperte o "P" com um dedo da outra mão.

Caso você tenha feito tudo corretamente o TK 90X lhe imprimirá os dois caracteres em vermelho que aparecem na tecla P e na tela da T.V. poderá ver agora:

PRINT"

À estes dois caracteres ou "pinguinhos" chama-se de *aspas*.

Agora tecle PRINT "Maravilhoso".

O que, não consegue fazer isto?

Também, não expliquei como fazer para escrever uma letra maiúscula e depois ter o resto com minúsculas....

Aí vai, o procedimento para se obter isto.

Em primeiro lugar você deve fazer o seu TK 90X entrar no modo C (já disse que C é a inicial de "capitals" que em inglês significa maiúsculas).

Mantendo apertada **CAPS SHIFT** pressione a letra M e veja como ela sai impressa como maiúscula.

Caso você queira permanecer neste modo por muito tempo basta apertar simultaneamente as teclas **CAPS SHIFT** e **CAPS LOCK** e aí todas as letras serão maiúsculas.

O cursor ficará no modo C até que você saia do mesmo apertando novamente e simultaneamente **CAPS SHIFT** e **CAPS LOCK**.

Repetindo, escreva

**PRINT** "Maravilhoso" **ENTER**

e a palavra "Maravilhoso" aparece na tela porém *sem aspas* e sem a *palavra chave PRINT*.

Já dá para concluir que se você quiser que o TK 90X use palavras basta que as mesmas apareçam entre os sinais de aspas e após o comando PRINT.

Isto é válido para letras e números.

Apenas para constatar a sua superioridade sobre o TK 90X mande o mesmo imprimir algumas operações tais como

**PRINT** "3+4=9" **ENTER**

**PRINT** "10-5=4" **ENTER**

Uê, uê, uê!!!!!!!

O meu, digo o seu, TK 90X não sabe nem somar e nem subtrair?

Resp.: O seu TK 90X fez o que você mandou ele fazer. Ele não somou 3+4 e tão pouco diminuiu 5 de 10. Ele apenas imprimiu o que você mandou ele imprimir.

Note que "-", "+" e "=" são os caracteres vermelhos nas letras J, K e L respectivamente e para obter os mesmos você deve manter pressionada a tecla **SYMBOL SHIFT** enquanto aperta cada uma destas teclas.

Tente agora entrar com a seguinte mensagem:

PRINT "A guerra nas estrelas é o maior perigo da atualidade."

É evidente, que para entrar com esta mensagem, no final de tudo você deve apertar a tecla **ENTER** ou seja logo após o segundo conjunto de aspas.

Note que para se obter os espaços em branco entre as diversas palavras você deve apertar a tecla **SPACE** (espaço).

Caso você cometa algum erro enquanto estiver teclando "pode passar a borracha" no mesmo e para tanto basta manter abaixada a tecla **CAPS SHIFT** e apertar a tecla 0 (zero) que tem a palavra **DELETE** impressa em branco acima da mesma.

Uma tradução para o DELETE pode ser *eliminar* ou *apague*.

Desta forma com esta operação *apaga-se* o caractere imediatamente à esquerda do cursor.

Você já percebeu que o TK 90X mantém a linha de baixo para mensagens?

Quando você tecla uma instrução para o TK 90X ela sempre aparece nesta parte da tela.

Também quando o TK 90X informa que está tu do 0 Executado 0:1 o mesmo sai lá embaixo...

Esta parte da tela é chamada de espaço para o trabalho.

O TK 90X usa o resto da tela para fazer o que você mandar que ele faça.

## Capítulo 4

### Fazendo operações matemáticas no TK 90X

#### Recordando

- PRINT " " → Diz ao TK 90X para imprimir na tela da T.V. tudo o que estiver dentro das aspas. As aspas não irão aparecer na tela.
- SYMBOL SHIFT** → Se você mantém esta tecla pressionada enquanto aperta uma outra tecla, obterá o caractere, símbolo ou palavra-chave que aparece em vermelho nesta tecla. Por exemplo, **SYMBOL SHIFT** e tecla "Z" lhe dará os dois pontos ":".
- CAPS SHIFT** → Se você mantém esta tecla abaixada enquanto aperta uma tecla que contém uma letra, a mesma será impressa na forma maiúscula na tela. Caso você utilize **CAPS SHIFT** com uma tecla que contém um número, obterá a função que aparece em branco acima da tecla numérica.
- DELETE** → Pode-se apagar ou eliminar um caractere introduzido erradamente mantendo abaixada a tecla **CAPS SHIFT** enquanto você aperta a tecla  $\emptyset$  isto é abaixa-se as duas simultaneamente. Com esta ação apaga-se o caractere imediatamente a esquerda do cursor.



Você seguiu as minhas recomendações até agora e não estás mais na estaca zero.

Porém o seu aprendizado deve aumentar em velocidade e para tanto debes seguir a seguinte recomendação: quando estiveres junto a mim (TK 90X) deves estar tranquilo(a) relaxado(a) e sem nenhum tipo de temor, visto que, a não ser que me maltrates fisicamente por mais absurdos que possas escrever não irás me machucar.

Eu (TK 90X) irei auxiliando-o(a) continuamente indicando os seus erros e apresentando-lhe os resultados.

Você pode aumentar a velocidade de leitura do livro e ir tão depressa quanto queiras com a condição de que não titubeies em voltar para trás quando achares isto necessário.

Eu sei que você não sabe nada e por esta razão estou indo pouco a pouco, porém agora vamos um pouco mais rápido e você deve se preparar para um esforço um pouco maior.

Somas, subtrações, multiplicações, divisões e exponenciações serão importantes para cálculos matemáticos, estatísticos, físico e por que não dizer até para os jogos, gráficos e para a música.

Somar, subtrair, multiplicar, dividir e elevar é algo muito simples para o seu TK 90X.

Símbolo	Significado para o TK 90X
=	me ordenam deixar "algo" IGUAL a "algo"
+	me ordenam SOMAR
-	me ordenam SUBTRAIR
*	me ordenam MULTIPLICAR
/	me ordenam DIVIDIR
↑	me ordenam ELEVAR

Tabela 1

Comece fazendo

PRINT 3+4

e em seguida aperte o .

No topo da tela aparece a resposta 7 e como você percebe o seu TK 90X sabe somar.

Note que no Capítulo 3 deu impressão que ele não sabia fazer isto...

Você pode mandar o TK 90X fazer uma soma bem maior do que esta.

Teclé

PRINT 10+15+20+25+30+35+9999

Assim que você apertar  o seu TK 90X lhe dá imediatamente a resposta

em muito menos que *um segundo*.

Vai ser rápido assim lá nos...

Você pode subtrair, dividir, multiplicar, elevar.

Na Tabela 1 você já notou que para indicar a multiplicação usa-se o asterisco "\*" no lugar do clássico "X", para a divisão usa-se a barra "/" e para a exponenciação (elevação a uma certa potência) a flecha para cima "↑".

Teste a habilidade do seu TK 90X efetuando os seguintes cálculos.

	Resposta
PRINT 4*3.14159 →	12.56636
PRINT 13/4.278 →	3.0388032
PRINT 5392*640 / 13 →	265452.31
PRINT 4984*328 - 55555/3.141 →	1617065
PRINT 13↑2 - 13*13 →	1.1920929E-7
PRINT 2↑4 + 2↑3 + 2↑2 + 2↑1 →	30

Não esqueça que em cada caso para sair o resultado indicado a direita você deve teclar o .

Cômodo e muito fácil, não é?

A ordem de prioridade com que o TK 90X realiza as operações depende do operador que intervêm.

OPERAÇÃO	OPERADOR	OBSERVAÇÃO
exponenciação	↑	prioridade máxima
multiplicação e divisão	* /	
soma e subtração	+ -	prioridade mínima

Tabela 2

Para operadores de mesma prioridade, o TK 90X executará as operações da esquerda para a direita.

Nestas condições quando temos:

4984\*328 - 55555 / 3.141

o TK 90X faz a multiplicação (4984 \* 328) depois a divisão (55555/3.141) e finalmente subtrai este resultado do primeiro.

As operações dentro dos parênteses se efetuam primeiro, seguindo dentro dos mesmos a ordem de prioridade indicada na Tabela 2.

Caso haja vários níveis de parênteses, os mesmos são abertos de dentro para fora.

Assim para efetuar o cálculo

$$\frac{2,38^{1,35} + 7 \times 3,35}{4,48 \times 325 - 777} \rightarrow \text{escrito na forma clássica}$$

Basta escrever:

PRINT (2.38 ↑ 1.35 + 7 \* 3.35) / (4.48 \* 325 - 777)

Tecla **ENTER** e deve obter .039284065

Com o treininho para sentir a destreza do TK 90X vamos efetuar os seguintes cálculos:

a)  $\frac{7 \times 5^{0,75} + 13}{2 \times 3,48 - 4}$

b)  $\frac{4,75 + \frac{3,98 \times 7,47 - 19}{3^{0,72} - 5,28}}{10,32 + \frac{9,28}{4,43} - 5,25^{0,21}}$

c)  $4,53 + \frac{2,28}{4,53^{0,25}} + \frac{1,28}{2,77^{0,31}} + \frac{0,73}{1,77^{0,18}} + 1$

Para obter os resultados, basta escrever

a) PRINT (7\*5 ↑ .75 + 13) / (2\*3.48 - 4) **ENTER**

b) PRINT (4.75+(3.98\*7.47-19)/(3↑.72-5.28))/(10.32+9.28/4.43 - 5.25 ↑ .21) **ENTER**

c) PRINT 4.53+2.28/4.53↑.25+1.28/2.77↑.31+.73/(1.77↑.18+1) **ENTER**

Você deve obter as seguintes respostas:

- 12.299294
- 0.11453544
- 7.3724276

Não esqueça então que:

- "+" → O sinal de "mais" diz ao TK 90X para adicionar o próximo número ao anterior.
- "-" → O sinal de "menos" diz ao TK 90X para subtrair o número seguinte do imediatamente anterior.
- "\*" → O asterisco (ou a "estrelinha") ou o sinal de multiplicação ("vezes") diz ao TK 90X para multiplicar dois números. Não use para a multiplicação o "antigo" X pois o TK 90X não entende isto como multiplicação.
- "/" → A barra é o sinal de divisão para o TK 90X. Ele não entende o símbolo ":" como sendo indicador da operação de divisão.
- "↑" → Este é o sinal para a operação de elevar a alguma potência (inteira ou decimal). É importante que na elevação a base sempre seja positiva.
- "=" → Este é o símbolo para exprimir a IGUALDADE.
- "( )" → O que está entre parênteses é sempre efetuado em primeiro lugar. A abertura de vários níveis de parênteses é sempre de dentro para fora.

$$\frac{747^3 - 528^2}{3456} ?$$



## Capítulo 5

### Obrigado TK 90X pela sua memória!!!

Você sabe que o TK 90X pode lembrar das coisas?

Você pode pensar na memória do TK 90X como sendo constituída de milhares de caixas vazias.

Em cada uma destas caixas de memória o TK 90X pode guardar ou armazenar um número ou uma palavra que você queira que ele lembre.

Suponha que você tem 27 camisas e 13 pares de sapatos.

Olhe no teclado e ache o comando LET na tecla L.

Tecla agora:

```
LET camisas = 27
```

Observe que *não aparecem as aspas* em torno da palavra "camisas".

A razão disto é que a palavra "camisas" está sendo usada como *rótulo* para uma caixa da memória e não como alguma coisa que o TK 90X tenha que imprimir (PRINT).

Agora aperte ENTER.

O que aconteceu?

Não muito, com exceção de que a sua última linha desapareceu.



Porém a mensagem Ø Executado Ø:1 está de volta e isto significa que o TK 9ØX fez o que você pediu para ele fazer.

Você pode verificar isto, porém teclando agora

```
PRINT camisas ( ENTER )
```

Agora o número 27 aparece impresso no topo da tela da T.V. e a esquerda.

Como foi que o TK 9ØX fez isto?

Bem, inicialmente ele achou uma caixa da memória rotulada com "CAMISAS" e "olhou" dentro da mesma.

Aí ele encontrou o número 27 e "copiou" o mesmo na tela deixando ainda o número 27 nesta caixa de memória.

Da mesma forma você pode ensinar o TK 9ØX a lembrar o número de pares de sapatos que você tem.

Escreva

```
LET sapatos = 13 ( ENTER )
```

O TK 9ØX sabe agora quantas camisas e pares de sapatos você possui.

Para que você tenha certeza disto basta teclar e entrar (ENTER) com as seguintes linhas:

```
PRINT camisas ( ENTER )
```

```
PRINT sapatos ( ENTER )
```

e note que os números 27 e 13 aparecem na tela.

Para que o TK 9ØX possa somar os artigos de vestuário que você possui basta escrever

```
LET artigos = camisas + sapatos ( ENTER )
```

Ø Executado Ø:1 diz o seu TK 9ØX não é?

Bem, agora teclando

```
PRINT artigos ( ENTER )
```

e você verá na tela a resposta 4Ø.

### Explicando melhor

O nome real para estas caixas de memória é *variável numérica*.

Este nome nasceu na língua dos "computadores" e significa que você pode variar ou modificar o número aí armazenado tantas vezes quantas você queira e como queira.

A sua liberdade junto com o TK 9ØX é quase indefinida.

Não existem na realidade milhares destas caixas dentro do TK 9ØX, porém esta é uma forma bem simples para pensar (não compreender) como é a memória do seu *super-micro*.

Você agora não pode esquecer a função do importante comando **LET** cuja forma geral é

```
LET variável = qualquer número
```

e que diz ao TK 9ØX para fazer com que uma letra ou uma palavra ou ainda uma letra seguida por um número (a qual chamamos de *variável*) se torne igual a algum número.

Por exemplo

```
{ LET galinhas = 37 ( ENTER )
  LET perus = 6 ( ENTER )
  LET patos = 16 ( ENTER )
```

fará com que o TK 9ØX armazene o número 37 na caixa de memória com o rótulo "galinhas", o número 6 no rótulo ou seja na variável numérica "perus" e 16 na variável numérica "patos".

Por outro lado

```
LET aves = galinhas + perus + patos ( ENTER )
```

farã com que o TK 90X guarde na caixa da memória com o rótulo "aves" o número total de animais que neste caso é igual a 59.

Para que este número apareça na tela basta agora você teclar

PRINT aves (ENTER)

O nome de uma variável numérica pode ser qualquer combinação de letras, dígitos (0 até 9) e espaços desde que o primeiro caractere seja sempre uma letra.

Assim

{ X  
x  
y2  
CUSTO ALTO  
Delta

são nomes válidos para variáveis.

Cuidado que uma variável não pode ser representada por 3X, 5A, etc.

O TK 90X também não distingue entre letras minúsculas e maiúsculas no nome de uma variável e ignora também os espaços em branco.

Desta maneira para o TK 90X

{ Delta  
DELTA  
delta  
Del ta

é o nome para uma mesma variável numérica

A eletrônica do TK 90X está essencialmente constituída por um microprocessador modelo Z80, que cumpre a função da unidade central de processamento (u.c.p.).

Este microprocessador é muito comum também em outros microcomputadores.

Além disto tem-se os "chips" de memória.

Entre estes deve-se distinguir os que "memorizam" a informação de forma permanente e saem gravados da fábrica, de acordo com as instruções do projeto da Microdigital (memória ROM - Read Only Memory) e aqueles cuja capacidade de memória desaparece quando se corta a corrente elétrica (memória RAM - Random Access Memory).

A memória ROM contém muitas rotinas especiais escritas em linguagem de máquina.

Toda vez que o TK 90X está ligado, o microprocessador Z80 executa uma ou outras das seguintes rotinas

- Permite entrar ou editar um programa.
- Executar o seu programa escrito em BASIC. A ROM contém o conjunto de "regras" que o Z80 usa para interpretar cada comando ou função em BASIC e desta forma efetuar o que você escreveu.
- Sentir se alguma das teclas foi pressionada.
- Imprimir na tela ou na impressora as informações.
- Salvar ou seja armazenar em uma fita cassete o seu programa ou ainda carregar da fita cassete para a memória do TK 90X.
- Contém os padrões ou modelos de pontos usados para formar os caracteres na tela da T.V. ou na impressora.

Mais uma vez insisto que toda esta informação está fixa no chip ROM quando ele é produzido.

Você pode facilmente ler o que existe no mesmo porém não pode alterar qualquer parte da informação guardada na ROM.

A memória RAM por outro lado é bem mais flexível.

Além de permitir que se leia o que está armazenado na memória RAM você (ou o microprocessador Z80) pode alterar aquilo que está aí, escrevendo novas informações.

A única desvantagem desta memória é que ela é *volátil* ou seja o conteúdo da mesma é perdida se for interrompida a corrente elétrica.

A RAM é dividida em várias áreas e entre elas destacam-se:

- Área para o arquivo da tela e atributos.
- Área para o programa em BASIC ou seja onde é guardado o seu programa.
- Área para as variáveis, usada para guardar os dados variáveis para o seu programa.

Uma das coisas que atíça a curiosidade dos (das) noviços(as) em computação é a forma como é medida a memória do TK 90X.

Aí é que vem a dúvida sobre o que são bytes, bits e K !?!?!?!?

A memória do TK 90X é organizada de tal forma que se possa seguir o microprocessador, que sendo um dispositivo binário "sente-se feliz" quando pode trabalhar com *números com potências de 2*.

Também o chip processador necessita que a memória seja organizada de uma forma uniforme.

Vamos agora pensar na memória do TK 90X como uma linha bem longa de caixas idênticas.

As caixas são numeradas consecutivamente, de forma que o microprocessador pode usar este *número* ou *endereço* para escolher uma particular caixa.

Em cada posição (isto é em cada "caixa"), está um número binário formado por 8 "bits" ou dígitos binários os quais podem ser "0" ou "1".

Um grupo de 8 bits é chamado de 1 byte.

Existem 256 (2 elevado a 8) arranjos com repetição de 2 elementos ("0" e "1") atribuídos a 8 posições de forma que pode-se pensar em cada posição da memória como contendo um número inteiro, na base dez, no intervalo de 0 até 255.

O TK 90X pode, naturalmente, manipular números fora deste intervalo, assim como as frações e para fazer isto utiliza um grupo de 5 bytes.

No total o microprocessador Z80 do TK 90X pode manipular 65536 ( $2^{16}$ ) diferentes posições da memória.

O primeiro quarto destas posições é tomado pela memória ROM e dependendo do modelo as restantes posições são parcialmente ou completamente ocupadas pela memória RAM.

Uma forma simplificada de se referir ao tamanho da memória é utilizar a abreviatura "K" para representar 1024 ( $2^{10} = 1024$ ) bytes.

Desta forma o TK 90X pode manipular 64 bytes de memória, 16 K da qual é ROM e as restantes 48K podem ser de memória RAM.

Note que  $16 \times 1024 = 16384$

$$48 \times 1024 = \frac{49152}{65536}$$

Observação Importante - O.I.

BASE DOIS	BASE DEZ
00000000	0
00000001	1
00000010	2
00000011	3
00000100	4
00000101	5
.....	...
11111111	255



Tabela 3

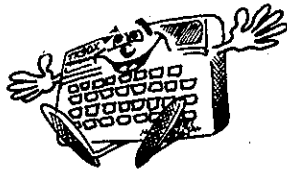
Para perceber isto veja quanto é  
 $(11011011)_2$

na base 10.

A forma para obter esta conversão é:

$$\begin{array}{cccccccc}
 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\
 & \swarrow & \swarrow & \swarrow & \swarrow & \swarrow & \swarrow & \swarrow & \swarrow \\
 (11011011)_2 & = & 1x2^7 & + & 1x2^6 & + & 0x2^5 & + & 1x2^4 & + & 1x2^3 & + & 0x2^2 & + & 1x2^1 & + & 1x2^0 \\
 & = & 128 & + & 64 & + & 0 & + & 16 & + & 8 & + & 0 & + & 2 & + & 1 & = & (219)_{10}
 \end{array}$$

Mostre que  $(11111111)_2 = (255)_{10}$ .



## Capítulo 6

### Lembrando das letras e das cadeias de letras

O TK 90X pode também lembrar de palavras e de sentenças.

Ele tem caixas especiais para armazenar seqüências alfabéticas ou alfanuméricas.

Porém para usar uma caixa especial para uma cadeia ou seqüência de caracteres (STRING) você precisa utilizar um rótulo todo especial.

Ele precisa *ser uma única letra* seguida pelo caractere cifrão (ou dollar) "\$".

Assim, caso você queira que o TK 90X lembre um nome pode chegar a este fim escrevendo:

```
LET n$ = "Sacha" ( ENTER )
```

**O.I.** - Tecler o seu próprio nome no lugar de "Sacha" que é o diminutivo em russo do nome do meu filho ca gula porém não esqueça de apresentá-lo entre aspas...

Para verificar se o TK 90X lembra ainda este (ou o seu nome) basta teclar

```
PRINT n$ ( ENTER )
```

O TK 90X pode também lembrar de uma sentença completa.

Tente agora

```
LET s$ = "As negociações serão um complexo de ques
tões relativas ao espaço e armas nucleares"
( ENTER )
```

Agora tecle

```
PRINT s$ ( ENTER )
```

e veja como sai impresso na parte de cima da tela a frase: "As negociações serão um complexo de questões relativas ao espaço e armas nucleares".

Caso você use a letra maiúscula "S" no lugar da minúscula "s" será que a mensagem é ainda a mesma.

Para verificar tecle:

```
PRINT S$ ( ENTER )
```

O que aconteceu?

Como você pode ver o TK 90X não reconhece a diferença entre rótulos com letras maiúsculas e aqueles com letras minúsculas.

Qualquer uma das 26 letras do alfabeto pode ser usada para rótulos ou seja para as variáveis STRING.

Desta maneira, no lugar de n\$, para guardar um nome você pode usar tanto A\$ como Z\$.

É entretanto uma boa idéia escolher como inicial para a variável STRING uma letra que tenha alguma associação com o que existe nesta caixa.

Uma coisa que é muito importante você entender (e não esquecer) é que uma caixa de memória pode armazenar uma STRING de cada vez.

Desta forma se você escrever:

```
LET n$ = "Joana" ( ENTER )
```

e aí então você colocar um outro nome nesta caixa

```
LET n$ = "Cristina" ( ENTER )
```

quando você tecla

```
PRINT n$
```

o seu TK 90X irá imprimir o último nome ou seja "Cristina".

É, para o TK 90X é bem fácil!!!!

Ele sempre só lembra da última e a imediatamente anterior é totalmente esquecida...

Q.I. - No jargão do "computadorês", a palavra para um grupo de caracteres que aparecem entre as aspas é STRING.

Uma variável pode armazenar uma STRING que pode ser uma coleção de caracteres alfabéticos ou gráficos ou ainda outros tipos de símbolos existentes no teclado do TK 90X.

No TK 90X uma STRING pode ter qualquer comprimento desde *nada* até uma STRING bem comprida cujo comprimento tem como limitação somente a quantidade de memória disponível.

Uma STRING com *nada* entre os sinais de aspas ("" ) é chamada de *STRING vazia*.

Você verá no final deste livro nos programas aplicativos, por incrível que possa parecer, a grande utilidade das STRINGS vazias.

Aí vai por exemplo uma forma para indicar ao TK 90X como esquecer totalmente algo

```
LET n$ = "" ( ENTER )
```

Portanto, a nova instrução que você conhece agora é:

```
RESUMINDO: LET x$ = "qualquer coisa"
```

Caso você utilize um rótulo consistindo *unicamente* de uma letra seguida pelo cifrão "\$" o TK 90X irá armazenar o que segue entre as aspas em tipo especial de caixa de memória.

Este tipo de variável é chamado de variável `STRING`.

Contrariamente as variáveis numéricas (que contêm somente números) as variáveis `STRING` podem conter qualquer tipo de informação incluindo números, palavras, caracteres gráficos e símbolos especiais.

Não esqueça que o TK 90X trata as letras maiúsculas de forma idêntica as minúsculas de forma que `X$` tem o mesmo significado para o seu super-micro que `x$`.



Tecele então o "mini-mini" exemplo

```
10 x$ = "Sergio"
20 PRINT x$
30 PRINT X$
```

Ao executar (**RUN**) sairá:

```
Sergio
Sergio
```

## Capítulo 7

### Finalmente escrevendo o seu primeiro programa!!!

Até agora o TK 90X tem feito tudo que você lhe pediu imediatamente.

Cada vez que você quèria que o TK 90X fizesse algo *nov*o, tinha que teclar uma nova instrução.

A esta forma de utilizar o TK 90X chama-se *modo imediato*.

Mas o seu TK 90X pode fazer coisa muito melhor!!!

Ele pode lembrar longas listas de instruções; as quais chamaremos de *PROGRAMA*.

Você ainda lembra quais eram os itens no Capítulo 5 que compunham o seu vestuário?

Inicialmente, você teclou os dois tipos que lhe interessavam (camisas e sapatos) e depois efetuou a soma dos mesmos.

Para transformar as instruções do *modo imediato* para o modo programado e desta forma constituir um *programa* tudo que você deve fazer é acrescentar um *número de linha* na frente de cada instrução.

Os números de linha dizem ao TK 90X a ordem na qual você quer que as instruções sejam executadas.

Antes de teclar um novo programa, você deve apertar a tecla marcada com "A", no qual está escrito em branco a palavra-chave **NEW**.

Aí aperte **ENTER**.

Com isto, limpa-se a memória do TK 90X e se evita que você tenha que tirar o "plug" da tomada a toda hora.

Tecele agora o seu *primeiro programa*

### PROGRAMA 1

10 LET camisas = 27

Você não esqueceu que caso cometa algum erro basta apertar **CAPS SHIFT** e 0 para corrigir a última posição imediatamente a esquerda do cursor.

Depois de ter apertado **ENTER**, você notará que esta linha irá reaparecer no topo da tela.

Continuando a teclar o programa entre com as linhas restantes

20 LET sapatos = 13 ( **ENTER** )

30 LET artigos = camisas + sapatos ( **ENTER** )

40 PRINT artigos ( **ENTER** )

Até agora não aconteceu nada, exceto que cada linha aparece na sua ordem correta no topo da tela.

Aperce agora a tecla "R" na qual em branco está escrita a palavra chave **RUN**.

Aí, a seguir aperte o **ENTER**.

Caso você tenha teclado tudo corretamente, aparecerá o número 40 no topo da tela.

Explicações adicionais sobre a entrada do programa e as dificuldades que podem ocorrer no decurso do mesmo

Comumente, quando você pressiona a tecla **ENTER** após ter teclado uma linha do programa o TK 90X faz com que a tela da T.V. fique branca por um momento.

Então ele escreve a linha no topo da tela e imprime o **K** piscando no espaço de trabalho na parte baixa e a esquerda da tela.

O **K** significa que o TK 90X aceitou a linha do programa e está esperando que você tecele uma palavra chave (tal como **RUN**) ou um outro número de linha do programa.

Ocasionalmente (assim espero...) você poderá se ver frente a situação na qual o TK 90X recusa-se a fazer isto.

Caso você olhe cuidadosamente, provavelmente verá um sinal de interrogação piscando "**?**" em algum ponto da linha que há pouco teclou e que não quer subir pois aí existe algum tipo de erro.

Por exemplo, se você esqueceu de teclar o sinal de igual "=" na linha 30, a sua linha terá o seguinte aspecto.

30 LET artigos camisas + sapatos

Quando você apertar **ENTER**, o TK 90X provavelmente imprimirá um ponto de interrogação onde ele acha que você cometeu um erro, isto é aparecerá:

30 LET artigos camisas? + sapatos

Esta é uma característica extremamente útil do BASIC do seu TK 90X.

O TK 90X verifica cada linha que você teclou antes de aceitar a mesma quanto a *erros de sintaxe*.

Para corrigir este erro utilize **CAPS SHIFT** e 0 (zero) para apagar tudo até chegar ao ponto de interrogação.

Então você pode rebater todo o resto da linha corretamente.

30 LET artigos = camisas + sapatos

Agora, quando você pressionar a teclar **ENTER** a linha corrigida irá até o topo ocupando o seu lugar correto.

Um outro problema com o qual você pode se defrontar é aquele no qual a linha parece que foi aceita porém nada aparece no topo da tela.

Ao invés disto, você pode ler mensagem como

0 Executado 0:1

no espaço de trabalho.

Isto provavelmente significa que você esqueceu de teclar o número da linha e o seu TK 90X pensa que você mandou uma instrução em modo imediato.

Não se preocupe e aperte logo o **ENTER** ou tra vez que as linhas do seu programa reaparecerão novamente na tela.

Rebata a linha que se perdeu e fique tranquilo(a)...

Espero que no seu caso particular apareça no canto baixo da esquerda a mensagem

0 Executado 40 : 1

com a qual o seu TK 90X indica que tudo correu bem!!

Terminando este capítulo não esqueça que:

**Programa** →

À toda série de instruções que você quer que sejam executadas em uma particular ordem chama-se de *programa*. A cada instrução é dado um número de linha.

Normalmente, no lugar de numerar as linhas com 1,2,3,4, etc.,... você deve numerá-las de 10 em 10 (10,20,30,40, etc.) de forma que você tenha espaço para inserir linhas adicionais caso isto seja necessário.

**NEW** →

Com este comando limpa-se a memória do TK 90X de qualquer programa anterior.

Tecler sempre o **NEW** antes de teclar um novo programa, porém esteja certo de que o programa antigo tenha sido armazenado em uma fita se ele é realmente importante para você,

**RUN** →

Diz ao TK 90X para executar as instruções começando com aquela de menor número de linha no seu programa. Portanto a ordem **RUN** seguida claro da executiva **ENTER** fará com que o TK 90X entenda que o seu desejo é que ele (TK 90X) "corra" (RUN) o programa ou que é a mesma coisa, ponha o mesmo em movimento ou em exploração.



## Capítulo 8

### Aprendendo a editar e corrigir erros

Existe uma forma bem simples de corrigir linhas de programa e diferente da utilizada até agora.

Pratique bastante no sistema de edição (EDIT) e logo você será muito mais rápido que o vento deslocando-se como um "superman" para todos os pontos da tela.

Inicialmente tecle o seu segundo e curto programa.

#### PROGRAMA 2

```
1Ø LET galinhas = 37      ( ENTER )
2Ø LET perus = 6          ( ENTER )
3Ø LET patos = 16         ( ENTER )
4Ø LET aves = galinhas+perus+patos ( ENTER )
5Ø PRINT aves             ( ENTER )
```

Imagine, por um momento, que você contou ou adquiriu um novo peru e quer aumentar "perus" na linha 2Ø para 7.

Caso você já tenha executado o programa pressione **ENTER** para obter a listagem do mesmo de volta.

Agora olhe para as teclas 5,6,7 e 8.

Você conseguiu notar as flechas brancas que existem acima destas teclas?

Mantenha pressionada a tecla **CAPS SHIFT** e aperte a tecla "6" que tem impressa acima a *flecha* (ou seta) *para baixo*.

A ponta de uma seta ou ponteiro (>) irá aparecer na listagem logo após o primeiro número de linha (no nosso caso particular o número 10). Chama-se a este ponteiro de *cursor da linha de programa*.

Aperte de novo **CAPS SHIFT** e "6" outra vez para levar o cursor da linha de programa para baixo ou seja para a linha 20.

Agora, mantenha abaixada a tecla **CAPS SHIFT** e aperte a tecla "1" que tem escrita acima de si a palavra **EDIT**.

Isto fará com que uma cópia da linha apareça no espaço de trabalho.

20 LET perus = 6

Mantenha ainda a tecla **CAPS SHIFT** abaixada e neste momento aperte a tecla "8" (flecha para a direita) várias vezes até que o cursor L esteja no fim da linha.

20 LET perus = 6 **L**

Agora é que chegou a hora de usar **CAPS SHIFT** e 0 para apagar (DELETE) o 6 e então teclar o 7.

Aperte o **ENTER** e a linha corrigida subirá para a sua posição correta ou seja

20 LET perus = 7

O cursor da linha de programa (>) estará novamente na linha 20 para que você não esqueça que esta foi a última linha teclada.

Execute (RUN) o programa e então aperte o **ENTER** para ter a listagem completa do programa novamente.

Use a *flecha para baixo* (como explicado há pouco) para levar o cursor da linha do programa para baixo e neste instante para a linha 50.

Então, edite a linha 50 embaixo, no espaço de trabalho.

50 PRINT aves

Neste momento, use a tecla da flecha para a esquerda (**CAPS SHIFT** e tecla "5") para mover o cursor uma posição para a esquerda.

5 **K** 0 PRINT aves

Utilize **CAPS SHIFT** e 0 para apagar (DELETE) o "5" e teclar o "6" para mudar a linha 50 para 60.

Use agora a flecha para a direita (não esqueça da **CAPS SHIFT**) para mover o cursor para o espaço após o PRINT.

60 PRINT **L** aves

Teclre agora

"Aves = **L**";

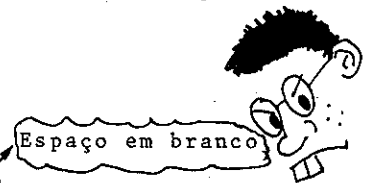
para tornar a nova linha 60 com o seguinte conteúdo:

60 PRINT "Aves = **L**"; aves

Veja o significado deste ponto e vírgula (;) no Capítulo 9

Esteja certo que incluiu ambos os conjuntos de aspas e o símbolo (vermelho) de ponto e vírgula";" que você encontrará sobre a tecla 0.

Quando você apertar o **ENTER**, a linha modificada entrará na listagem acima, mas a antiga



linha 50 também aparecerá.

Caso você queira eliminar totalmente a linha 50 tecla simplesmente:

50 ( **ENTER** )

Agora execute ( **RUN** ) o programa revisto.

Pratique o mais que puder a edição usando as teclas com as flechas até que você possa editar linhas, fazer mudanças, inserções e eliminações rapidamente e sem nenhuma dificuldade...

Não esqueça então o conceito básico deste capítulo que é o **EDIT**.

O sistema de edição (EDIT) lhe permite fazer mudanças em um programa existente corrigindo particulares linhas no lugar de rebater trechos do mesmo.



## Capítulo 9

### O comando INPUT, um novo e futuro grande amigo seu

Se a palavra-chave PRINT foi até este ponto o seu comando favorito, a próxima palavra-chave a entrar neste rol é

**INPUT**

Com o auxílio deste comando você poderá economizar muito do seu tempo dedicado a teclar.

Caso você queira informar algo ao TK 90X, como por exemplo o número de camisas, sapatos, perús etc. que possui, não será mais necessário fazê-lo através de instruções do tipo:

30 LET patos = 16

Aliás, esta forma de definir o valor de uma variável é bastante inadequada quando se quer modificar alguma informação pertinente à mesma.

Para mudar o número de "patos", você deve editar (EDIT) ou rebater esta linha do programa.

É neste capítulo, que ensinarei a você, uma forma mais simples de fazer isto usando a palavra-chave (branca).

**INPUT**

a qual você encontra na tecla "I".

Tecla agora o seu terceiro (e ainda curtiinho) programa.

Não esqueça de teclar o **ENTER** após cada linha.

Aliás já considero isto como um hábito adotado e daqui para frente poucas vezes iremos repetir isto no final de cada linha.

### PROGRAMA 3

```
1Ø PRINT "Qual é o seu nome?"
2Ø INPUT n$
3Ø PRINT "Como vai você?"
4Ø PRINT n$
```

Antes de executar ( **RUN** ) o programa tente adivinhar o que fará o mesmo.

Aí vai a explicação de como ele opera.

Linha 1Ø → Você usou o seu "velho amigo" PRINT para imprimir uma pergunta na tela da T.V.

Linha 2Ø → A linha com o comando INPUT faz com que o TK 9ØX pare e espere que você teclasse algo.

O cifrão "\$" após a letra "n" diz ao TK 9ØX que você deseja que ele aguarde por uma STRING de caracteres (neste caso, o seu nome), a qual é armazenada na caixa da memória que foi rotulada n\$.

O TK 9ØX ficará esperando para sempre ou até que você aperte a tecla **ENTER** para informar que terminou de teclar.

Quando você terminar de teclar e pressionar a tecla **ENTER**, o TK 9ØX deslocar-se-á para a próxima linha do programa - linha 3Ø.

Linha 3Ø → Aí, simplesmente se imprime a mensagem "Como vai você".

Linha 4Ø → Indica ao TK 9ØX para olhar na caixa de memória que se chamou de n\$ e imprimir tudo que estiver contido dentro da mesma, na tela da T.V.

Execute ( **RUN** ) o programa mais uma vez para verificar se entendeu tudo o que foi dito há pouco.



### Melhorando o seu terceiro programa

Agora você já sabe como o programa trabalha e pode-se fazer algumas mudanças no mesmo com o intuito de melhorá-lo.

Inicialmente, se você quer experimentar o programa com um(uma) amigo(a) que nada saiba sobre micros, ele (ou ela) logo ficará gamado(a) pelo TK 9ØX.

O seu (a sua) amigo(a) pode não perceber que o TK 9ØX quer que você teclasse o seu nome.

E ainda mais, certamente não perceberá que você precisa apertar a tecla **ENTER** para fazer com que o programa continue.

Em outras palavras o programa não é "muito amigável" ou seja não é auto instrutivo ou suficiente para pessoas que não entendem de micros ou mesmo que não tenham conhecimento prévio do programa.

Desta forma no lugar de perguntar simplesmente

"Qual é o seu nome?".

seria muito mais explicativo e educado para o(a) usuário(a) dizer:

"Por favor, teclasse o seu nome e depois aperte a tecla **ENTER**".

Realmente, você pode fazer algo mais com o intuito de diminuir o seu tempo de digitação no futuro.

Há uma forma especial de uso do comando INPUT no qual se pergunta e ao mesmo tempo se entra com aresposta como por exemplo:

```
INPUT "Qual é o seu nome?"
```

INPUT pode também receber a resposta, colocar a mesma na caixa da memória e tudo isto na mesma linha do programa como:

```
10 INPUT "Qual é o seu nome?";n$
```

Você notou a presença do ponto e vírgula (;) antes do n\$?

É muito importante incluir o ponto e vírgula (;) pois caso contrário o TK 90X se recusará a aceitar esta linha do programa.

O ponto e vírgula é o símbolo vermelho que está na tecla 0.

Destá forma torne o seu programa auto explicativo e otimizado com a seguinte linha 10 (nova):

```
10 INPUT "Por favor, tecla o seu nome e depois aperte a tecla ENTER";n$
```

Depois de ter feito isto, é óbvio que deve-se eliminar a antiga linha 20 o que se consegue entrando com:

```
20 ( ENTER )
```

Olhe agora no seu programa e observe que a linha 20 desapareceu.

Execute (RUN) novamente o programa.

Você notou alguma diferença?

No lugar de se imprimir a pergunta no topo da tela, o comando INPUT imprime a mensagem no espaço de trabalho, na parte de baixo da tela.

Existe uma outra coisa que você pode melhorar.

Se o seu nome é digamos igual ao meu, Victor, não seria mais elegante no lugar de imprimir

```
Como vai você
Victor
```

imprimir

```
Como vai você Victor
```

tudo numa mesma linha?

Você pode mandar o TK 90X fazer isto de uma forma bem simples, acrescentando no final da linha 30 um ponto e vírgula (;).

Os pontos e vírgula (;) tem um significado muito especial para o TK 90X.

Eles informam ao TK 90X que a impressão deve começar imediatamente após e não numa nova linha.

Edite (EDIT) a linha 30 para que no final tenha o seguinte aspecto:

```
30 PRINT "Como vai você";
```

Quando você executar (RUN) o terceiro programa, já revisto, verá no topo da tela algo assim:

```
Como vai vocêVictor
```

Melhor, mas ainda não totalmente o que se queria...

Você precisa de algum espaço entre as palavras "você" e "Victor".

Edite a linha 30 e inclua dois espaços em branco após a palavra "você" e antes das aspas

```
30 PRINT "Como vai você";
```

2 espaços em branco

Quando você estiver fazendo isto, por que não tentar acrescentar o "n\$" da linha 40 no final da linha 30?

Isto já funcionou na linha 10 com o INPUT de forma que tente agora.

```
30 PRINT "Como vai você";n$
```

Aperte **ENTER** e a linha se deslocará para cima substituindo a antiga linha 30 na sua listagem.

Antes de executar (RUN) o programa revisito pela segunda vez, você precisa eliminar a linha 40 e para tanto basta teclar

```
40 ( ENTER )
```

Agora execute o programa.

Está como se queria e com um programa bem menor, não é?

Neste capítulo as novas instruções foram:

**INPUT variável numérica** → Diz ao TK 90X para esperar até que um número seja teclado. O comando INPUT está na tecla I

**INPUT variável STRING** → Diz ao TK 90X para esperar até que entre uma letra, palavra ou uma cadeia (STRING) de caracteres.

**INPUT "mensagem";n\$** → Permite que você envie uma mensagem (entre aspas) e após isto espera a entrada de uma variável STRING que deve aparecer separada da mensagem por um ponto e vírgula (;).



Veja agora se entende os programas 1, 2, e 3 do Capítulo 23.

Vá até 14 e depois de entendê-los volte ao Capítulo 10.

## Capítulo 10

### Divertindo-se com a programação

Agora você já não é tão neófito(a) em programação e desta forma vamos ao quarto programa, mais longo e totalmente comentado.

Se o terceiro programa (Capítulo 9) não está mais no seu TK 90X tecler

#### PROGRAMA 4

```
10 INPUT "Por favor, tecler o seu nome e depois aperte a tecla ENTER";n$
```

```
30 PRINT "Como vai você";n$
```

Acrescente agora as seguintes linhas:

```
40 PRINT
```

```
50 PRINT "O meu nome é TK 90X"
```

```
60 PRINT
```

```
70 PRINT "Diga-me";n$
```

```
80 INPUT "Qual é a sua idade?";idade
```

```
90 PRINT
```

```
100 PRINT "Cêus-"; idade; "anos de idade!"
```

```
110 PRINT "Você é bem mais velho do que eu!"
```

Até este ponto convém ressaltar alguns detalhes.

Inicialmente, esteja seguro de incluir to-

dos os pontos e vírgula (;) ou o seu TK 9ØX se recu-  
sarã terminantemente a aceitar estas linhas.

Não esqueça de incluir os espaços em bran-  
co (□) onde eu os coloquei, isto é, dentro dos tre-  
chos entre aspas.

Por exemplo, na linha 8Ø os dois espaços  
em branco são colocados para separar a pergunta do  
valor numérico, correspondente a sua idade.

Também na linha 1ØØ incluímos vários espa-  
ços em branco para se ter uma saída simples deler!!

Não se preocupe, se uma linha é tão com-  
prida que vai para a outra linha da tela, como é o  
caso por exemplo da linha 1Ø.

Fica por conta do TK 9ØX interromper uma  
linha onde ele achar necessário.

Você provavelmente não vai ter muita difi-  
culdade em achar a maior parte dos sinais de pon-  
tuação, todos em vermelho.

Não esqueça que para isto se deve utilizar  
a tecla (em vermelho) **SYMBOL SHIFT** e aí se obtém  
qualquer coisa que esteja em vermelho.

Tome cuidado para não confundir a vírgula  
(,) que está na tecla "N" com o apóstrofe "'" que  
está na tecla "7".

Antes de executar (RUN) o programa você po-  
de advinhar qual é a função do comando PRINT que  
aparece nas linhas 4Ø, 6Ø e 9Ø?

Bem, execute (RUN) o programa para ver se  
você estava certo!

Você achou que um PRINT sem nada após o  
mesmo significa que deve-se imprimir (PRINT) uma li-  
nha vazia deixando um espaço entre as outras linhas  
e fazendo com isto que o que se exhibe na tela seja

mais simples de ler?

Parabêns bicho (gatinha)!!!!

Caso seja possível, mantenha este programa  
na memória do TK 9ØX, pronto para ser incrementa-  
do!!!

Os(as) habituados(as) com microcomputado-  
res referem-se a uma lista de linhas de programas  
como sendo uma *listagem*.

No TK 9ØX você pode listar (LIST) qualquer  
parte de um programa, desde que o programa tenha si-  
do parado após apertar-se duas vezes a tecla **ENTER**  
ou pressionando a tecla "K", a qual tem a palavra-  
chave **LIST** impressa sobre a mesma, sempre que o  
cursor **K** estiver na tela da T.V.

No momento, o seu programa é ainda bem cur-  
to, de forma que você pode ver todo o programa lis-  
tado na tela ao mesmo tempo.

Mais tarde você quererã ver partes de pro-  
gramas compridos, tão compridos que sã dá para ver  
trechos dos mesmos.

Tente isto, teclando para o último (quarto)  
programa:

 LIST 5Ø ( **ENTER** )  Oi a observação sobre  
o **ENTER** para entrar  
com algo no TK 9ØX

Com este comando o TK 9ØX irá mostrar para  
você todas as linhas da 5Ø em diante.

Teste o que acontece se você teclar

LIST ( **ENTER** )

sem um número de linha e então com diversos números.

Nã já vimos que o TK 9ØX é muito bom com  
as operações de somar, subtrair, multiplicar, divi-

dir e elevar a uma certa potência (Capítulo 4).

Para tornar, os programas mais interessantes, vou incluir à partir deste ponto algumas linhas que contenham fórmulas matemáticas bem simples.

Liste (LIST) seu último programa para ver se existem quaisquer números com os quais pode-se obrigar o TK 90X a operar.

É isto aí, na linha 80 você entrou com a sua idade e é com ela que vamos brincar um pouco.

Não esqueça que qualquer coisa que esteja em uma caixa de variável numérica pode ser tratada como um número comum ou seja ser usado em operações algébricas.

Não se amofine e nem esquite a preciosa "cuquinha" antes do tempo.

Por enquanto tecle as seguintes linhas que devem ser incorporadas ao seu quarto e último programa:

```
120 PRINT
130 LET velho = 70 - idade
140 PRINT "Você está ciente que somente daqui "; velho;
150 PRINT "anos, será septuagenário?"
```

Na linha 130 tudo o que se faz é subtrair a sua idade de 70 e colocar a resposta ou o resultado na caixa de memória numérica denominada "velho".

Desta forma se a sua idade atual é 15 ter-se-á a variável "velho" com valor 55.

Execute ( RUN ) agora o programa e tente compreender melhor como funciona o mesmo.

Execute o mesmo várias vezes, usando diferentes idades.

O que ocorre se a sua avó, que tem hoje 78 anos, entrar com a idade dela?

Neste caso sairá uma mensagem com idade negativa e para que não se interprete isto erradamente convém modificar a linha 30 pelo menos para:

```
130 LET velho = 80 - idade.
```

Porém a aritmética até agora pedida ao TK 90X é muito simples e por isto vamos acrescentar mais algumas linhas ao programa.

O ano tem 365 dias e a cada quatro anos tem-se um ano bissexto ou seja o ano em que o mês de fevereiro tem 29 dias e portanto neste ano tem-se 366 dias.

Assim em média um ano tem 365,25 dias.

Portanto se você multiplicar a sua idade por 365,25 você obterá o número de dias que já viveu (aproximadamente).

Acrescente portanto as seguintes linhas:

```
160 PRINT
170 LET dias = idade * 365.25
180 PRINT "Você já viveu "; dias; " dias."
```

O ponto decimal (a nossa clássica vírgula decimal) é o caractere em vermelho que está na tela "M".

Como um dia tem 24 horas, você pode também querer saber quantas horas já viveu.

Acrescente as seguintes linhas:

```
190 PRINT
200 LET horas = dias * 24
210 PRINT "Se cada hora que passa na nossa vida é importante saiba que:"
220 PRINT "Você já viveu "; horas; " horas."
```

Dá para expressar o que você já viveu em minutos? E em segundos?



É claro que sim desde que você introduza que toda hora corresponde a 60 minutos e cada minuto corresponde a 60 segundos.

Complete então o programa com as seguintes linhas:

```
230 PRINT
240 LET minutos = 60 * horas
250 PRINT "Em minutos parece que a sua vida é
bem grande!!!"
260 PRINT "Saiba que você já viveu [ ]"; minutos;
"[ ] minutos"
270 PRINT
280 LET segundos = 60 * minutos
290 PRINT "Em segundos dá impressão que a sua
vida é infinita!!!"
300 PRINT "Se você tem mais que 3 anos de idade
a que você já viveu em segundos será apresen
tado na notação exponencial"
310 PRINT "A sua vida medida em segundos é: [ ]";
segundos
```

Agora execute (RUN) o programa, para ver que tipo de respostas obtêm!

O.I. - Você ainda lembra como transformar um número na forma exponencial em um número na forma normal?

Dado que, para este quarto programa não devemos ter nenhum número negativo basta você deslocar o ponto decimal para a direita tantas posições quanto é o valor do número logo após o E.

Dessa forma se o número na forma exponencial for

4.73364 E+8 segundos

você precisa deslocar o ponto decimal 8 posições para a direita acrescentando neste caso 3 "zeros" extra.

O equivalente ao número 4.73364 E+8 na forma comum é 473 364 000 segundos ou seja 473 milhões e 364 mil segundos e que equivale a uma idade atual de 15 anos.

Neste capítulo a instrução nova que você não pode esquecer é

**LIST** → Diz ao TK 90X para listar (LIST) e exibir o seu programa começando com a primeira linha do programa.

**LIST número** → Por exemplo se no nosso último programa escrevermos LIST 160 isto ordena ao TK 90X que liste somente as linhas de programa começando no número teclado, neste caso 160.



# Capítulo 11

## Guardando e carregando os seus programas

Agora que você já teclou alguns programas (todos não muito longos...) seria uma boa idéia que o seu TK 90X "permitisse" guardar os mesmos em fitas cassete, desde que o(a) caro(a) leitor(leitora) tenha um gravador adequado.

Caso você não tenha um gravador pode pular este capítulo até que adquira um.

Compre logo um, pois você precisará dele.

Muitos programas são tão longos que será necessário batê-los em diversas etapas.

Caso isto ocorra é fundamental poder guardá-los (SAVE) por exemplo em uma fita cassete e depois carregá-los (LOAD) para o TK 90X na próxima vez possibilitando a continuação da "teclagem" a partir do ponto da última parada.

O TK 90X guarda programas transformando as informações que você lhe manda para a sua memória em sons que podem ser gravados em fitas cassete comuns através do microfone (MIC) do seu gravador.

O TK 90X pode então verificar (VERIFY) se tudo foi gravado adequadamente antes que você desligue o mesmo.

Caso você queira usar o programa novamente ligue o TK 90X, volte a fita cassete e carregue (LOAD) o programa para a memória do TK 90X, porém

desta vez conecte o mesmo ao áudio (EAR socket) do seu gravador.

Em resumo, você guarda (SAVE) o programa através do MIC e carrega (LOAD) o programa através do EAR (veja a Figura 1).

É muito fácil fazer isto, só dependendo do ajuste do volume, que aliás pode lhe trazer no início, um pouco de "dor de cabeça" até que você se acostume com isto.

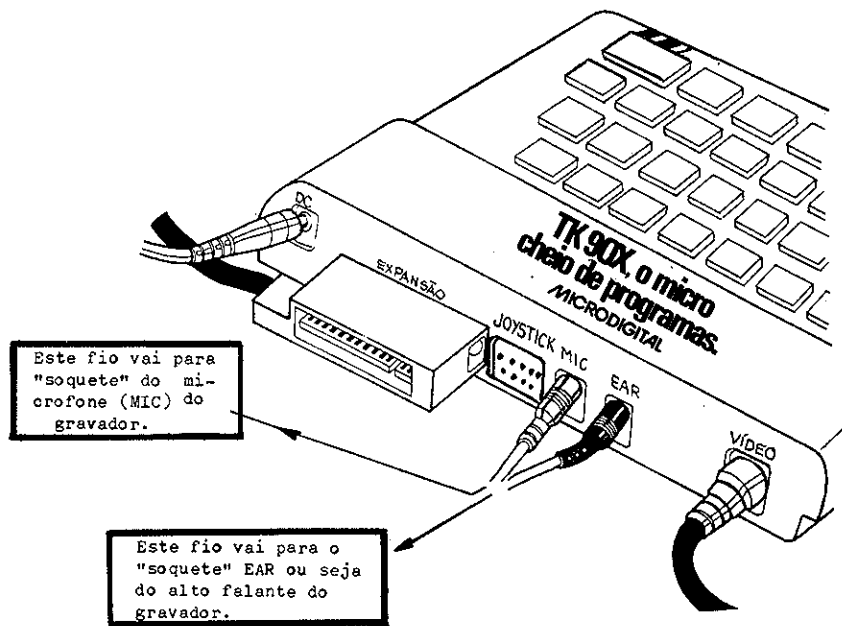


Figura 1

Aqui pode surgir a pergunta: "Que tipo de gravador devo ter?".

E aí vai a incrível resposta: "O melhor gravador para o TK 90X é o gravador de preço módico, portátil e do tipo "mono".

Os gravadores mais incrementados poderão servir mas não são tão eficientes.

O gravador que você deve adquirir deve ter no mínimo duas pequenas entradas, (3,5 mm) ou tomadas marcadas com MIC (ou AUX IN) e EAR (ou LS pois em inglês "loudspeaker" significa alto falante ou ainda AUX OUT).

Se você estiver comprando um gravador para utilizar especialmente com o TK 90X leve em conta estimado(a) leitor(leitora) as seguintes observações.

- 1) Diga ao vendedor que você quer um gravador para ser usado com o TK 90X.  
Pergunte-lhe se poderá trocá-lo ou devolvê-lo caso ele não funcione adequadamente com o TK 90X.  
Alguns gravadores podem não funcionar bem em conexão com o seu TK 90X, mas isto só dá para saber depois que você efetuar a compra...
- 2) É interessante que o seu (futuro) gravador tenha um contador ou medidor de fita.  
Isto pode ser muito útil se você estiver tentando achar um particular programa entre vários que podem ter sido gravados na mesma fita.
- 3) É mais econômico ter um gravador que possa ser ligado na rede elétrica, porém se funcionar com bateria isto lhe permitirá as vezes um rendimento melhor a um custo bem maior a não ser que você possa recarregar as baterias.
- 4) Tenha certeza que existem as embocaduras EAR e MIC como foi explicado acima.
- 5) Verifique se o gravador pode carregar (LOAD) tanto os programas a venda no comércio como aqueles que você próprio irá gravar e depois carregar.

### Guardando programas na fita

Em primeiro lugar tenha certeza que existe algum programa na memória do TK 90X apertando a tecla **ENTER**.

Que memória que você tem, hein!?!?!

Caso não exista nenhum programa, tecla algum ou melhor copie o mais simples dos que já aprendeu até agora.

Antes que você guarde (SAVE) o seu programa é necessário que o mesmo tenha um *nome curto*.

Suponha que você decidiu chamar o seu programa de "teste".

Em primeiro lugar efetue a conexão entre o MIC do gravador e o MIC do TK 90X.

Em seguida aperte a tecla S que tem sobre si a palavra-chave SAVE e tecla o nome do seu programa entre aspas ou seja

SAVE "teste" (**ENTER**)

Lembre que o gravador deve estar ligado no modo de gravação (PLAY/RECORD). Em alguns gravadores basta apertar o RECORD e por isto veja também as instruções do seu aparelho.

Depois de ter teclado **ENTER** aparecerão nas bordas da tela faixas horizontais e sons agudos.

Ao primeiro sinal se produz a gravação do nome do programa, e faixas de cor vermelha e azul são vistas durante uns 5 segundos.

Depois disto quando as faixas são mais juntas e em cores azul e amarela o próprio programa está sendo gravado.

Espera até que apareça a mensagem



**Ø Executado Ø : 1**

Pare o gravador pois o programa está gravado!!!

Você pode não ter se convencido disto então tecla

**VERIFY "teste"**

VERIFY é uma palavra em vermelho abaixo da tecla **R**.

Mantenha abaixada **CAPS SHIFT** enquanto você aperta **SYMBOL SHIFT**.

O cursor piscante **K** muda para o faiscante **E** no modo estendido.

Novamente pressione a tecla **CAPS SHIFT** e mantendo-a abaixada aperte a tecla R.

Desta forma você obterá a palavra **VERIFY**.

Tecla o nome do seu programa, entre aspas, da mesma maneira que você fez anteriormente com o comando **SAVE**.

Caso tenha esquecido o nome do seu programa tecla:

**VERIFY""**

Agora encaixe o fio na embocadura EAR no gravador e no TK 90X.

Volte a fita ao ponto de partida e coloque o volume a três quartos do máximo.

Aperte a tecla PLAY do gravador e em seguida aperte a tecla ENTER do TK 90X.

Se realmente você armazenou (SAVE) o seu programa com sucesso terá uma tela branca seguida de um bordo faiscante.

Então aparecerão as mesmas faixas de cor vermelha e azul no bordo, que você viu quando estava armazenando o seu programa, seguidas de faixas

estreitas de cor azul e amarela.

Depois das faixas de cor azul e amarela, o nome do programa será impresso no topo da tela com a palavra "Programa" em frente do mesmo ou seja

```
Programa: teste
```

Finalmente quando se chegar ao fim deve aparecer a mensagem

```
Ø Executado Ø : 1
```

indicando que você tem na fita uma boa cópia do seu programa e com toda confiança pode desligar o TK 90X.



### Carregando programas

Não se aborreça se levar algum tempo para surgir o primeiro SAVE com sucesso.

Nem tudo é fácil na vida, no início...

Mas o TK 90X pode fazer mais coisas com o gravador e entre elas esta: *carregar programas*.

Para carregar (LOAD) um programa para a memória do TK 90X ligue o gravador da mesma forma que você fez há pouco com o VERIFY ou melhor para que se possa ler o programa da fita siga os seguintes passos:

- 1) Ajustar o volume do gravador entre 6 e 8
- 2) Colocar no agudo máximo no caso de se ter um gravador mono
- 3) Acionar o comando LOAD"" ou LOAD "teste"
- 4) Pressionar a tecla PLAY do gravador
- 5) Pressionar a tecla ENTER do TK 90X
- 6) Esperar pela mensagem

```
Ø Executado Ø : 1
```

Não esqueça que se não souber o nome do programa basta escrever LOAD"" (sem nenhum espaço em branco entre as aspas) porém neste caso o TK 90X carregará o primeiro programa que vier na fita.

Ele também imprime o nome do programa na tela para que você tenha um lembrete.

As novas instruções que apareceram neste capítulo são:

**SAVE "teste"** → Diz ao TK 90X para guardar o que tem na sua memória em uma fita cassete com a denominação "teste".  
O nome do programa pode ter até um máximo de dez caracteres.

**VERIFY "teste"** → Diz ao TK 90X para verificar se o programa foi armazenado corretamente.

**LOAD "teste"** → Diz ao TK 90X para carregar para a sua memória da fita cassete um programa de nome "teste".

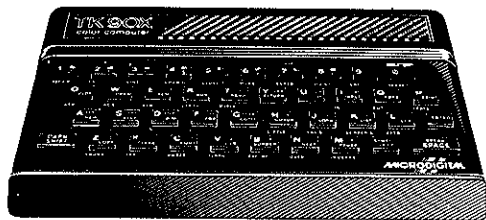
**LOAD""** → Diz ao TK 90X para carregar o primeiro programa que encontrar na fita.  
Entre as aspas não deve haver nada. Qualquer uma dessas instruções pode ser incluída em um trecho do seu programa. Assim geralmente no fim de um programa pode-se escrever

```
888 SAVE "teste":VERIFY "teste"
e com um GOTO 888 estaremos guardando e verificando o programa.
```

Este procedimento é muito útil para programas longos que introduzimos no TK 90X em etapas e queremos armazenar e verificar as cópias dos mesmos.



**O. I.** - Convém neste ponto o(a) atento(a) leitor(lei-  
tora) ler o Capítulo 23 do seu Manual de Ope-  
ração do TK 90X pois lá encontrará informa-  
ções complementares sobre o uso correto de  
SAVE, VERIFY e LOAD.



**TK 90X, o micro  
cheio de programas.**  
MICRODIGITAL

## Capítulo 12

### Como ter uma possibilidade ligada a um acontecimento aleatório

Logo, logo você começará a aprender como se pode programar jogos porém antes é importante que saiba como se pode escolher ou obter um *número aleatório*.

Um número aleatório é o tipo do número que você pode obter ao lançar um dado.

Em um dado honesto, qualquer número entre 1 e 6 tem a mesma possibilidade ou melhor, probabilidade de sair, cada vez que se lança o mesmo.

Vejamos como o TK 90X pode imitar um dado.

Caso você olhe para a tecla "T", verá a "sigla" RND impressa em azul acima da mesma.

Você pode acionar esta instrução apertando as teclas **CAPS SHIFT** e **SYMBOL SHIFT** junto, para obter o cursor **E** piscando.

Aí sim, deve-se apertar a tecla "T" e ter-se-á a função que gera números aleatórios - função RND de Random (aleatório em inglês).

Tecele pois o seguinte mini-programa (*seu quinto programa*).

## PROGRAMA 5

1Ø PRINT RND

2Ø GOTO 1Ø

Quando você executar (RUN) este programa, verá números que o TK 9ØX acredita sejam números aleatórios.

Devido a instrução da linha 1Ø imprime-se um número aleatório entre 0 e 1.

Devido a instrução da linha 2Ø o programa volta para a linha 1Ø e imprime um outro número e um outro número e outro até que a tela fique cheia com 22 números aleatórios ocupando cada um uma linha.

Aí surge a mensagem

```
scroll?
```

no espaço de trabalho.

Caso você apertar qualquer tecla que não seja "N" (a qual significa "não" neste exemplo), `SPACE` ou `STOP`, a tela irá "rolar" como um anti-go pergaminho, e o TK 9ØX irá imprimir um novo conjunto de números aleatórios enchendo a tela e deixando você perplexo(a) e admirado(a)!!

Por causa do efeito de

```
2Ø GOTO 1Ø
```

envia-se o TK 9ØX a um laço (LOOP) sem fim como se fosse um cão tentando pegar o seu rabo.

O TK 9ØX dará voltas e voltas continuamente oferecendo-lhe em cada uma delas um número ininteligível...

A forma mais simples para parar a impressão destes números é apertar a tecla `SPACE` ou então a tecla "N".

Pessoas mais sofisticadas e que gostam de vida cheia de artifícios podem também usar a sequência `SYMBOL SHIFT` e "A" para parar o programa.

Isto produz a mensagem:

```
D BREAK - CONT repete 1Ø:1
```

de forma que pressione a tecla `CONT`

se você precisar - porém você terá também que apertar `ENTER` !

Depois que você parar um programa como este, se pressionar `ENTER` a listagem do seu programa será mostrada sem que haja a necessidade de teclar `LIST`.

Mas qual é a utilidade direta de frações decimais tão compridas, todas entre 0 e 1 como as que você acabou de ver?

Na realidade, não é muito grande, porém facilmente pode-se transformar estas frações decimais em números entre 1 e 6 para simular o que ocorre no lançamento de um dado.

Inicialmente multipliquemos a função RND por seis (6).

Tecla então uma linha nova [ou edite(EDIT) a antiga linha 1Ø].

```
1Ø PRINT RND *6
```

Quando você executar (RUN) este programa revisto os números aparecerão no intervalo de 0 (o qual você não quer) a 6 (o qual você quer mas que jamais será atingido e portanto exibido).

Além disto os números tem ainda muitas casas decimais.

Você pode se livrar desta indesejável parte decimal usando a instrução ou seja a função INT, impressa em azul acima da tecla "R".

Note que você precisa do modo estendido (modo E) ou ampliado para "chegar" até o INT de forma que use as teclas **CAPS SHIFT** e **SYMBOL SHIFT** junto ou seja ao mesmo tempo.

Modifique a linha 10 para

```
10 PRINT INT (RND * 6)
```

e não esqueça dos parênteses!!!!

Execute (RUN) outra vez o programa.

Tem um aspecto bem melhor, não é?

A função INT realmente nos livrou da incômoda parte decimal dos números porém resta ainda um problema os números que estão inscritos nas faces de um dado são 1,2,3,4,5 e 6 enquanto o que se exhibe na tela são os números 0,1,2,3,4 e 5.

Você é capaz de adivinhar como iremos ensinar o TK 90X a transformar 0 em 1 e o 5 em 6?

Aí vai a resposta:

```
10 PRINT INT(RND * 6) + 1
```

Por enquanto não é muito importante se você não está entendendo o que estou fazendo desde que compreenda que se vai obter os números que se quer.

Realmente `INT(RND*6)+1` produz o seguinte efeito:



- INT → elimina a parte fracionária dos números positivos
- RND → produz um número aleatório entre 0 e 1 sendo que o extremo 1 não está incluído.
- 6 → multiplica-se por 6 que é um número, uma unidade a mais que o valor da diferença entre o maior e o menor número que você quer (6-1=5)
- 1 → acrescenta-se o menor número que você quer.

A palavra-chave em branco na tecla T é

**RAND.**

Quando você apertar esta tecla, a palavra RAND será impressa na tela.

Isto é necessário para que o TK 90X leia números aleatórios de uma lista.

Quando o TK 90X é desligado, ao ser ligado novamente ele começa a ler esta lista sempre no mesmo lugar.

RAND diz ao TK 90X para ler os números aleatórios a partir de uma posição aleatória nesta lista, de forma que você "jamais" obterá a mesma série de números aleatórios!?!?!?

Existe uma coisa muito importante que você deve aprender neste ponto, antes por exemplo de participar do seu primeiro jogo no TK 90X.

Você já conhece o sinal de igual (=) conforme foi indicado na Tabela 2.

O TK 90X entende outros símbolos similares e que serão muito úteis daqui para frente.

Símbolo	Significado	Encontra-se na tecla
=	igual a	L
<	menor do que	R
>	maior do que	T
<=	menor do que ou igual a	Q
>=	maior do que ou igual a	E
<>	diferente ou não igual a	W

Tabela 4 - Operadores relacionais



Todos estes símbolos ou sinais estão desenhados com cor vermelha nas teclas indicadas na Tabela 4.

Caso não tivéssemos estes símbolos como é que poderíamos descartar todas as candidatas a miss Universo se as condições elementares fossem medir mais do que 1,70 metros e pesar menos que 60 quilogramas?

Na realidade todos estes símbolos aparecerão dentro das instruções IF-THEN como por exemplo:

```
IF chute < número THEN faça algo
```

onde se (IF) o seu "chute" for menor que um certo "número" então (THEN) deve ser feito algo.

Uma outra forma pode ser:

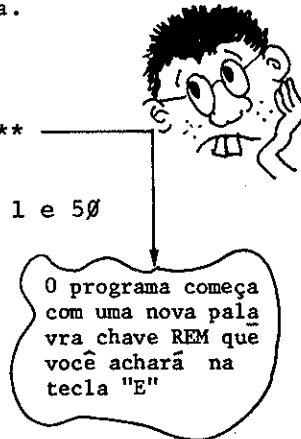
```
IF chute <> número THEN faça algo
```

que significa se (IF) o seu "chute" é diferente (ou não igual) ao "número", então (THEN) faça algo.

Falaremos já, já mais um pouco sobre o IF...THEN e muito no Capítulo 13 porém por enquanto comece a teclar o sexto programa.

### PROGRAMA 6

```
10 REM *** O JOGO DA ADVINHAÇÃO***
20 RAND → RANDOMIZE
30 PRINT "Advinhe o número entre 1 e 50
   que eu TK 90X escolhi"
40 LET número = INT (50*RND)+1
50 LET tentativa = 0
60 INPUT "Entre com o seu
   chute"; chute
70 LET tentativa = tentativa+1
```



```
80 PRINT "Na tentativa"; tentativa; "o seu
   chute foi"; chute
90 IF chute < número THEN PRINT "Escolheu um número
   menor"
100 IF chute > número THEN PRINT "Escolheu um número
   maior"
110 IF chute <> número THEN GOTO 60
120 PRINT "Parabéns, está correto!!!"
130 SOUND 3,7
140 CLS
150 GOTO 30
```



Execute o programa e então veja se você pode parar o mesmo!!!

Você vai sentir que o seu TK 90X as vezes se entusiasma tanto que é duro para parar.

Não adianta apertar a tecla **BREAK**.

A saída é pressionar a tecla **SYMBOL SHIFT** e a tecla "A" onde está o **STOP**

Deu para parar?

Uootimo!!!

Explicações detalhadas sobre o *sexto* programa.

Linha 10 → **REM** é uma palavra chave muito útil (está na tecla "E") pois permite que se possa incluir comentários (remarks em inglês) ou observações nos programas, tais como o título que foi introduzido.

O TK 90X ignora tudo que estiver na linha logo após o REM ou seja o conteúdo daquilo que vem após o REM em nada influencia a execução do programa.

Linha 20 → **RAND** diz ao TK 90X para escolher uma posição inicial aleatória dentro da sua lista de números aleatórios.

Linha 30 → Imprime uma mensagem no topo da tela sobre o jogo.

Linha 40 → Diz ao TK 90X para escolher um número aleatoriamente entre 1 e 50.

Linha 50 → Aí se inicializa a variável "tentativa" com a qual se vai contar quantas vezes você irá chutar até acertar no número que o TK 90X escolheu.

Linha 60 → Aí o TK 90X lhe pede para entrar com a sua adivinhação que será armazenada na variável "chute".

Linha 70 → Conta-se o número de tentativas acrescentando-se cada vez que se passa por aí uma unidade.

Linha 80 → Imprime-se o número da tentativa e o número escolhido por você (jogador ou jogadora).

Linha 90 → Imprime-se a mensagem somente se o seu "chute" for *menor do que* o número correto - *você está abaixo*.

Linha 100 → Imprime-se a mensagem somente se o seu "chute" for *maior do que* o número correto - *você está acima*.

Linha 110 → Diz ao TK 90X para ir de volta a linha 60 se o seu "chute" não é igual ao número correto.

Linha 120 → Caso o programa atinja este ponto sem que algo tenha sido impresso devido as linhas 90 ou 100 e sem ter sido enviado de volta pelo desvio incondicional GOTO 60, então finalmente você acertou. Como prêmio deve ver a mensagem "Parabéns, está correto!!!"

Linha 130 → Faz com que o TK 90X emita um som para comemorar o seu feito e com isto provocar uma pequena pausa, para que se tenha tempo de ler a última mensagem.

Falaremos adiante muito mais sobre o comando SOUND que é sempre seguido de dois números o primeiro indicando o comprimento de uma nota e o segundo (separado por vírgula do primeiro) a altura ou tom da mesma.

Mas isto será nos programas musicais.

Linha 140 → Com o comando CLS (está na tecla "V") manda-se limpar a tela e assim estar pronto para outro jogo. Em inglês "clear screen"

Linha 150 → Desvia incondicionalmente para a linha 30 para um novo jogo.

### PROGRAMA 7

Neste programa o seu TK 90X lhe dará o número e o seu logaritmo natural para cada número que você enviar ao TK 90X.

Caso o número introduzido seja maior que  $10^{21}$  ou seja  $1E+21$  manda-se parar o programa (linha 80).

Aí está o programa:

```

10 PRINT "Entre com o número"
20 INPUT x
30 IF x > 1E+21 THEN GOTO 80
40 PRINT "Número", "Logaritmo"
50 PRINT x, LN x / LN 10
60 PRINT
70 GOTO 10
80 STOP

```

Olhe a instrução IF-THEN fazendo o desvio quando o número que você introduzir for muito grande



### Explicações

1) A função LN número calcula o logaritmo natural ou seja na base  $e$  ( $e = 2,7182818$ ) de um "número" positivo.

Ocorre erro caso o número seja nulo ou negativo.

A função LN está em azul acima da tecla Z

Como queremos o logarítmo na base 10 (dez) utilizou-se a fórmula matemática

$$\log_{10} x = \frac{\log_e x}{\log_e 10}$$

2) A instrução STOP da linha 80 é para finalizar a execução de um programa.

**Q.I.** Caso você não saiba o que é um logarítmo consulte alguém que esteja cursando o 2º grau.

Muitas foram as novas informações deste capítulo lembre então que:

**GOTO** → Diz ao TK 90X para desviar (para frente ou para trás) para uma particular linha cujo número é indicado após o GOTO. Assim a palavra chave GOTO (está na tecla "G") é usada para modificar a ordem na qual as instruções do programa são executadas fazendo com que o TK 90X "pule" para uma outra linha. Para não esquecer o efeito do GOTO tecle o seguinte mini-programa



```
10 PRINT "TK 90X";
15 POKE 23692,0
20 GOTO 10
```

Este comando está na letra "O" e fará com que, o que se exhibe na tela, vá subindo para sempre sem lhe pedir permissão e evitando a mensagem "scroll?". Caso você não tenha entendido esta linha não se preocupe por enquanto. Trate-a como se fosse uma infalível fórmula mágica.

Para parar este "LOOP" sem fim a única forma sem desligar o TK 90X da tomada é apertando simultaneamente as teclas **CAPS SHIFT** e **BREAK**.

**RND** → Diz ao TK 90X para escolher um número aleatório entre 0 e 1 de uma lista pré definida. 0 RND está acima da tecla "T".

**RAND(omize)** → Diz ao TK 90X para escolher um número aleatório de um ponto aleatório desta lista. 0 RAND está na tecla "T".

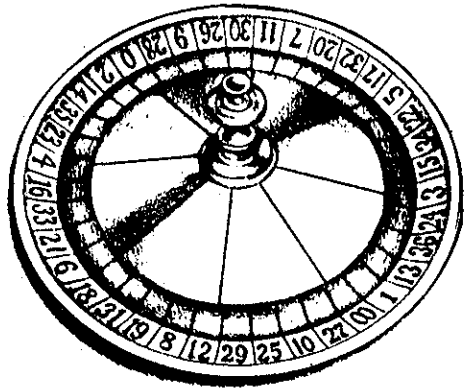
**INT** → A função INT diz ao TK 90X para cortar a parte de um número após o ponto decimal. Esta função está sobre a tecla "R".

Sobre o IF-THEN, o CLS e o SOUND falaremos muito adiante.



No momento veja se dá para entender o programa 4 do Capítulo 23.

Vá lá (GOTO) mas volte pois não é um desvio sem volta...



## Capítulo 13 Fazendo decisões com o auxílio do TK 90X

Uma das grandes vantagens que possui o TK 90X é a possibilidade de ser programado de tal forma que possa "tomar decisões".

Logo, logo você aprenderá como se pode programar o TK 90X para que o mesmo verifique se por exemplo a tecla que foi apertada é para mover uma nave para cima, para baixo, para a esquerda ou para a direita ou ainda se deve ser disparado o míssil da mesma.

Assim um trecho de um programa poderá ter o seguinte aspecto

```

IF (se) a tecla "5" for pressionada THEN (então) mova a nave para esquerda
IF (se) a tecla "8" for pressionada THEN (então) mova a nave para a direita
IF (se) a tecla "6" for pressionada THEN (então) mova a nave para baixo
IF (se) a tecla "7" for pressionada THEN (então) mova a nave para cima
IF (se) a tecla "D" for pressionada THEN (então) dispare o míssil
IF (se) o míssil atingiu o objetivo THEN (então) simule uma explosão e acrescente 100 pontos à sua contagem.

```

etc.

Em geral você programará o TK 9ØX para fazer decisões usando a instrução **IF...THEN**.

Como já comentei de "raspão" no Capítulo 12, o efeito do **IF...THEN** é dizer ao TK 9ØX que:

IF(se) algo é verdadeiro (como "é verdade que a tecla "5" foi pressionada) THEN (então) o TK 9ØX precisa fazer algo (como deslocar a nave para a esquerda).

É claro que você ainda não está apto para programar jogos espaciais e por isto o nosso *oitavo programa* usará muito a instrução **IF...THEN** e ao mesmo tempo o colocará em contato com os recursos coloridos do seu TK 9ØX.

Espero que você tenha uma T.V. colorida em casa pois caso contrário isto é numa T.V. em preto e branco o máximo que verás são os efeitos em diferentes tons de cinza.

Tecle por enquanto o seguinte

### PROGRAMA 8

```

1Ø CLS
2Ø PRINT "Você gostaria de mudar:"
3Ø PRINT "1 - a cor da borda (BORDER)"
4Ø PRINT "2 - a cor do papel (PAPER)"
5Ø PRINT "3 - a cor da tinta (INK)"
6Ø INPUT "Entre com 1,2 ou 3□"; escolha
7Ø IF escolha = 1 THEN GOTO 200
8Ø IF escolha = 2 THEN GOTO 300
9Ø IF escolha = 3 THEN GOTO 400
1ØØ PRINT "Ouça humanóide, o número precisa ser
1,2 ou 3! Não brinque comigo!"
11Ø PRINT "Não estou zangado, por favor tente de
novo..."

```

```

12Ø GOTO 2Ø
2ØØ REM A sua escolha recaiu sobre o 1 (BORDER)
21Ø CLS
22Ø INPUT "Qual é a cor que você quer (Ø-7)?□";
cor
23Ø BORDER Cor
24Ø PRINT "O número da cor da borda (BORDER) é:
□"; cor
25Ø PRINT
26Ø GOTO 2Ø
3ØØ REM A sua escolha recaiu sobre o 2 (PAPER)
31Ø CLS
32Ø INPUT "Qual é a cor que você quer (Ø-7);□";
cor
33Ø PAPER cor: CLS: PAPER cor
34Ø PRINT "O número da cor do papel PAPER cor:□"
cor
35Ø PRINT
36Ø GOTO 2Ø
4ØØ REM A sua escolha recaiu sobre o 3 (INK)
41Ø CLS
42Ø INPUT " Qual é a cor que você quer (Ø-7)?
□"; cor
43Ø INK cor
44Ø PRINT "O número da cor da tinta (INK) é:□";
cor
45Ø PRINT
46Ø GOTO 2Ø

```

Explicações detalhadas sobre o oitavo programa.

Linha 1Ø → Com esta instrução limpa-se a tela toda para se receber o novo programa.

Linha 2Ø a 6Ø → Espero que esta parte esteja totalmente clara para você, pelo que se explicou nos capítulos anteriores.

Caso isto não seja verdade ("true") volte imediatamente para o ponto em que tem dúvidas...

Linhas 70 a 90 → Em primeiro lugar a palavra chave (branca) IF está na tecla "U" de forma que só aperte a mesma após ter teclado o número de linha.

Já para obter o THEN você deve manter abaixada a tecla **SYMBOL SHIFT** enquanto aperta a tecla "G".

Note que sem apertar a **SYMBOL SHIFT** apertando a tecla "G" obtêm-se o GOTO.

Ao se usar o THEN o TK 90X automaticamente se prepara para reconhecer que a próxima tecla a ser pressionada corresponde a uma palavra chave (comando) e não uma simples letra.

Caso a sua "escolha" for igual a "1" vai-se para 200, caso seja igual a "2" vai-se para 300, caso for "3" vai-se para 400 e caso você entrar com um número diferente de 1,2 ou 3 o controle do programa é transferido para a linha de número 100.

Linhas 100 a 120 → Bem aí aparecem as mensagens de que você entrou com número-código errado e pede-se que você entre com a "escolha" novamente fazendo voltar para a linha 20 devido ao GOTO da linha 120.

Linhas 200,300 e 400 → Estas linhas não são estritamente necessárias porém é útil incluir uma instrução REM no início de cada trecho distinto de um programa para destacar para você o que realmente faz o seu TK 90X neste trecho.

No trecho 200 a 260 escolhe-se a cor da borda, mostra-se isto e volta-se a 20.

No trecho 300 a 360 escolhe-se a cor do papel, mostra-se isto e volta-se a 20.

No trecho 400 a 460 escolhe-se a cor da tinta, mostra-se isto e volta-se a 20

Linhas 210, 310 e 410 → Com o comando CLS limpa-se simplesmente tudo que há na tela de forma que você possa a partir deste ponto começar com uma tela nova e limpa.

Linhas 220, 320 e 420 → Utiliza-se o INPUT para parar o programa e desta forma você pode informar ao TK 90X o número da cor que você deseja. Ele então armazena este número na caixa da memória identificada pelo nome de variável "cor".

O TK 90X tem oito cores numeradas de 0 (preto) até o 7 (branco) de forma que "(0-7)?" é simplesmente o intervalo no qual estão todas as entradas *legais*.

As outras cores estão indicadas no teclado acima dos números 1,2,3,4,5 e 6.

Não esqueça que se você entrar com um número que não seja 0,1,2,3,4,5,6 ou 7 isto provoca uma entrada *ilegal*.

Ao "rodar" o programa dê uma entrada ilegal para ver o que acontece.

Linha 230 → O que você deve fazer aqui é apertar a palavra-chave (branca) BORDER que está na tecla "B".



Por favor não teclé este comando como um conjunto de letras separadas B,O,R,D,E,R. Vai dar erro, viu?!?!?

Aliás o mesmo se aplica a REM, CLS, PRINT, etc.

O seu TK 90X já lhe apresenta a vantagem da simplificação e por isto não procure o caminho da complicação...

Com BORDER define-se a cor que se quer para a borda, margem, beira, orla, fronteira ou simplesmente da área que faz o contorno da área de impressão.

Assim por exemplo com BORDER 6 se ordena ao TK 90X para fazer amarela esta região externa da tela.

Linhas 240, 340 e 440 → No caso da linha 240 imprime-se na tela a mensagem "o número da cor da borda (BORDER) é □".

O ponto e vírgula (;) e a palavra "cor" após a mensagem, diz ao TK 90X:

"imediatamente após o espaço em branco depois do "é" você deve imprimir o número que está contido na caixa de memória com o rótulo "cor" ".

É análogo o que ocorre nas linhas 340 (se bem que aí é para o PAPER) e 440 (aí para o INK).

Linhas 250, 350 e 450 → O PRINT sem nada depois dele faz com que o TK 90X deixe uma linha em branco. Como já disse anteriormente é apenas para fazer com que na tela da T.V. seja mais simples ler e identificar as respostas. Como você queira sentir a diferença, execute o programa removendo estas linhas do mesmo e depois execute novamente com todas estas linhas presentes.

Linhas 260, 360 e 460 → Com a instrução GOTO 20 informa-se ao TK 90X para voltar para a linha 20 do programa e começar tudo de novo.

A economia que você obtém com isto é que desta maneira não é necessário teclar toda vez o RUN para ver novamente uma outra cor

Linha 330 → Para se poder entrar com PAPER, cuja cor é vermelha você deve estar no modo estendido (modo E) e para tanto mantenha abaixada a tecla **CAPS SHIFT** enquanto aperta a tecla **SYMBOL SHIFT**.

O cursor que está piscando, deve mudar de **K** para **E**.

Com a tecla **CAPS SHIFT** abaixada outra vez você deve apertar a tecla "C" e aí surgirá escrito PAPER.

Se você esquecer de apertar novamente a tecla **CAPS SHIFT** e apertar somente a tecla "C" aparecerá o que está escrito em azul acima da tecla "C" ou seja LPRINT.

Caso isto ocorra, use o DELETE e então volte novamente ao modo estendido e tente de novo.

Até agora não demos grandes detalhes sobre a tela do seu aparelho de T.V. quando o mesmo está ligado ao TK 90X.

A tela da sua T.V. fica como se fosse um quadro negro no qual o TK 90X tem escrito as respostas às perguntas que tem sido feitas a ele.

Bem, como o TK 90X não tem olhos como você ele precisa de pontos de referência aos quais vai dirigir-se para poder imprimir bem na posição que lhe for indicada.

Com este fim o TK 90X considera a tela dividida em 24 linhas e 32 colunas, numeradas de 0 em diante a partir do canto superior da esquerda conforme está indicado na Figura 2.

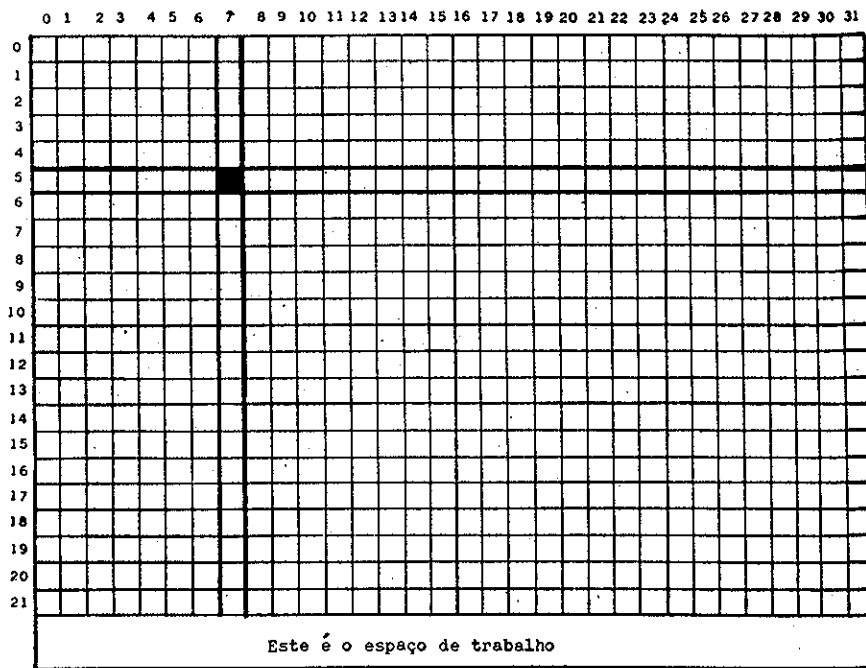


Figura 2

As linhas 22 e 23 estão reservadas para que o TK 90X possa comunicar-se contigo humanóide ou seja não estão disponíveis para você no que se refere a apresentar resultados e linhas de programa - este é o espaço de trabalho.

Portanto o TK 90X considera a tela do seu aparelho de T.V. dividida em um retângulo de 24 linhas e 32 colunas ficando desta forma claramente de finidas duas zonas na tela; a primeira que corresponde a zona onde se produzirão as impressões correspondentes às mensagens e informação em geral proveniente do TK 90X e a segunda que é todo o resto da tela que rodeia este retângulo.

É esta última zona que se pinta com o comando BORDER.

A palavra PAPER significa papel em inglês e é esta a instrução que determina, junto com o número que a segue, a cor que deve ter o retângulo onde são feitas as impressões.

Por exemplo com PAPER 2 se colorirá de vermelho este retângulo.

Quando você executar (RUN) o programa poderá notar que certas combinações de cores não funcionarão muito bem...

Por exemplo, se você tentar imprimir (PRINT) letras em preto em um fundo preto, é evidente que não conseguirá ler nada.







O.I. Não esqueça, que se você estiver em dificuldades pode sempre apertar a tecla **SYMBOL SHIFT** enquanto aperta também a tecla "A" para parar (STOP) o "problema".

Então tente teclar PAPER 7 (usando a tecla "C" no modo E) para que se tenha um fundo branco.

Finalmente apertando o **ENTER** e então LIST você deverá voltar a normalidade...

Linha 430 → Com esta instrução "ensina-se" ao TK 90X para mudar a cor da tinta (em inglês a palavra INK significa tinta) para o valor que está armazenado na caixa de memória com o rótulo "cor".

É com esta função (use **CAPS SHIFT**, **SYMBOL SHIFT** apertadas simultaneamente para entrar no modo E e depois **CAPS SHIFT** e a tecla "X" para poder escrever a instrução INK) que se faz com que os caracteres que aparecem na zona de impressão tenham a cor que você queira.

É evidente que não se deve ter PAPER (papel) e INK (tinta) com os números iguais após os mesmos.

Deve-se destacar que quando o seu TK 90X é ligado o BORDER e PAPER estão automaticamente na cor branca (número 7) e INK é preto (número 0).



O.I. - Caso você queira que toda a tela mude de cor, deve refazer a tela com uma instrução CLS antes da próxima instrução PRINT e desta forma acrescentar as seguintes linhas.

Caso você não queira isto, PAPER e INK influenciarão apenas na instrução PRINT atual.

Bem, as novas e importantes instruções que você não deve esquecer mais são as seguintes:

**IF...THEN** → Diz ao TK 90X para efetuar um teste que permita saber se uma condição é verdadeira e o que deve ser feito nesta situação.

**CLS** → Diz ao TK 90X para limpar a tela para o conjunto de cores.  
O comando CLS está na tecla "V".

**BORDER** → Diz ao TK 90X para "ajustar" a cor da borda ou seja da margem.  
Note que o BORDER está na tecla "B".

**PAPER** → Diz ao TK 90X para ajustar a cor do fundo ou seja a cor do "papel" que é o retângulo central da tela.

Para se obter o PAPER após a sequência **CAPS SHIFT**, **SYMBOL SHIFT**, **CAPS SHIFT** modo E deve-se apertar a tecla "C".

**INK** → Diz ao TK 90X para ajustar a cor do primeiro plano ou seja a cor da "tinta".

Para se obter o INK após a sequência **CAPS SHIFT**, **SYMBOL SHIFT**, **CAPS SHIFT** modo E deve-se apertar a tecla "X".

## Capítulo 14

### Utilizando subrotinas para inclusive obter economia de digitação



O oitavo programa ainda está na memória do TK 90X?

Espero ao menos que você tenha tido a boa idéia de tê-lo guardado numa fita.

Carregue-o e dê um LIST.

O que, nem está na memória e nem salvou o mesmo?

Péssimo!!!!

Aí não tem jeito vai ter que bater uma boa parte dele novamente.

Antes entretanto note que muitas linhas eram praticamente idênticas no oitavo programa ou seja

```
210 CLS
```

```
220 INPUT "Qual é a cor que você quer (0-7)?";
cor
```

e o mesmo acontece com (menos os números de linha)

```
310 CLS
```

```
320 INPUT "Qual é a cor que você quer (0-7)?";
cor
```

e ainda

```
410 CLS
```

```
420 INPUT "Qual é a cor que você quer (0-7)?";
cor
```

O TK 90X tem uma forma bem simples de imprimir estas duas linhas mais do que uma vez.

E aí, que pode ser usada uma *subrotina*.

### O que é uma subrotina?

Resp.: Uma subrotina é nada mais, nada menos que um pequeno programa dentro do próprio programa.

As linhas de 200 a 260 formam uma subrotina, o mesmo se pode dizer das linhas de 300 a 360 e também das linhas 400 até 460.

Após se desviar para uma ou outra dessas subrotinas (o que aliás foi decidido pela instrução IF...THEN), a mesma era executada segundo instruções como:

```
200 REM A sua escolha recaiu sobre o 1 (BORDER)
210 CLS
220 INPUT "Qual é a cor que você quer (0-7)?"; cor
230 BORDER cor
240 PRINT "O número da cor da borda (BORDER) é:
    "; cor
250 PRINT
260 GOTO 20
```



Usando o comando GOSUB

A instrução GOTO é muito boa quando usamos uma subrotina apenas uma vez.

Porém, se você estiver usando as mesmas linhas de programa várias vezes no programa como fizemos com as linhas 210 e 220 ou 310 e 320 ou ainda 410 e 420 o uso de uma subrotina que é chamada pelo comando GOSUB é preferível...

Se associamos a idéia de que um programa é uma rotina, será imediato concluir que uma subro-

tina é um programa que está dentro de uma rotina geral ou seja o *próprio programa*.

Em inglês, na suposição que estavam querendo mandar alguém (de preferência o TK 90X) a uma subrotina dirias, "goto subroutine" (vá para a subrotina) e para voltar da mesma falarias "return" (volte).

É foi pensando nisto que se criaram no BASIC as instruções GOSUB, para sair do programa principal, efetuar algum processo de trabalho na subrotina e RETURN, para voltar ao programa geral exatamente na linha que vem imediatamente após a linha do GOSUB.

Pode-se dizer que as "rotinas" GOSUB são quase idênticas às "rotinas" GOTO.

A grande e fundamental diferença é que o TK 90X lembra onde ele estava no programa principal e volta para a posição seguinte a esta quando no final da subrotina encontrar a instrução RETURN.

Para mostrar-lhe como isto funciona vamos criar uma nova (e primeira) subrotina para fazer o trabalho de limpar a tela (CLS) e perguntar-lhe qual é a cor que você deseja.

É na tecla "Y" que você encontra a palavra-chave (branca) RETURN.

Acrescente ao oitavo programa as seguintes linhas

```
500 REM Subrotina para a limpeza da tela e
    escolha da cor
510 CLS
520 INPUT "Qual é a cor que você quer (0-7)?"; cor
530 RETURN
```

Isto é tudo que você terá na subrotina.

Agora, para que tudo funcione direitinho elimine as linhas

210 ( ENTER )

310 ( ENTER )

410 ( ENTER )

que não são mais necessárias e modifique as linhas 220, 320 e 420 introduzindo a palavra-chave (branca) GOSUB que está na tecla "H" da seguinte forma:

220 GOSUB 500

320 GOSUB 500

420 GOSUB 500

Como um toque final, para evitar que você cometa alguma confusão para reconhecer este programa, que sem dúvida era bom ter gravado numa fita, coloque nele o título

5 REM \*\*\* CORES E SUBROTINA \*\*\*

Isto auxiliará você a lembrar que esta é uma versão otimizada do oitavo programa, no qual se tinha o contato com as cores do seu TK 90X.

Ah!Ah!Ah!

Ia esquecendo isto!!!

Execute (RUN) o programa "CORES E SUBROTINA" agora com a sua primeira subrotina.

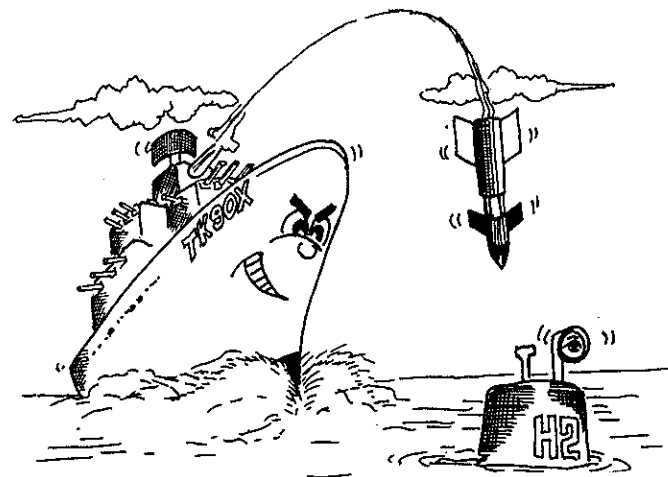
As novas instruções deste capítulo foram:

**GOSUB** → Diz ao TK 90X para lembrar a sua posição e ir (GO) até a subrotina (SUB).

Esta palavra-chave em cor branca está na tecla "H".

**RETURN** → Diz ao TK 90X para voltar a posição imediatamente seguinte àquela que ele lembra quando saiu devido ao comando GOSUB.

A palavra-chave RETURN, que tem cor branca, está na tecla "Y".



## Capítulo 15

### Movendo coisas na tela

Você já está acostumado a ver o seu TK 90X a imprimir coisas nos mais diferentes pontos da tela.

Porém se você quiser por exemplo elaborar programas para jogos eletrônicos precisarão saber como controlar onde as coisas aparecem na tela e principalmente saber movimentá-las.

Inicialmente você deve aprender como o TK 90X pode ser ensinado a imprimir em qualquer posição da tela.

Já lhe foi dito que a tela onde são feitas as impressões é constituída de 22 linhas numeradas de 0 (zero) no topo até a linha em baixo e logo acima do espaço para o trabalho cujo número é 21 (veja a Figura 2).

Cada uma destas linhas pode conter 32 caracteres ao longo da tela.

O espaço individual no qual cada letra, número ou símbolo pode ser impresso pode ser chamado de "retângulo".

Como as colunas são numeradas de 0 à esquerda, até 31 à direita da tela pode-se de forma única definir cada um dos  $32 \times 22 = 704$  retângulos onde se quer começar a imprimir.

Desta forma caso você queira que saia uma letra bem em cima e bem a esquerda da tela a sua posição seria:

linha 0, coluna 0

a qual o TK 90X entende como posição 0,0.

Se você quer imprimir um caractere no topo e no canto da direita da tela a sua posição seria:

linha 0, coluna 31

ou 0,31 de uma forma mais compacta.

O retângulo mais baixo e no canto da esquerda (imediatamente acima do espaço de trabalho) seria indicado por

linha 21, coluna 0 ou 21,0 de uma forma

mais compacta.

Como é que se especifica o canto baixo e da direita?

Resp.: 21,31.

Tecla agora a seguinte instrução

PRINT AT 4,10; "SOU O SEU TK 90X"

Você precisa usar a instrução (vermelha) AT, a qual está impressa sobre a tecla "I" e precisa também incluir a vírgula e o ponto e vírgula.

O comando que você deu há pouco ao TK 90X significa imprima (PRINT) a mensagem "SOU O SEU TK 90X" na linha número 4 e começando a impressão a partir da coluna número 10. (veja a Figura 3).

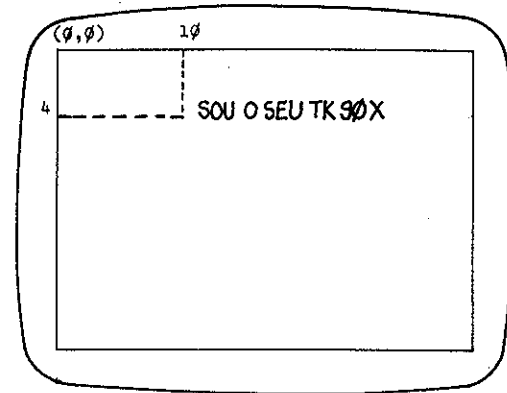


Figura 3

Vamos no nono programa fazer com que um nome se mova ao longo de uma linha e para tanto precisamos ensinar o TK 90X a imprimi-lo a partir de diferentes posições.

Tecla o programa e a seguir execute o mesmo.

Veja se compreende como o mesmo funciona e se identifica algumas incoerências...

#### PROGRAMA 9

```

10 REM *** MOVENDO UM NOME ***
20 LET linha = 11
30 LET coluna = 0
40 INPUT "Qual é o seu primeiro nome? "; n$
50 PRINT AT linha, coluna; n$
60 LET coluna = coluna + 1
70 GOTO 50

```

Você não acha que está tudo acontecendo um pouco depressa?

Concordas!!!

Que tal provocar um pequeno atraso?

Como é uma boa idéia vamos introduzir uma pausa musical com o auxílio do SOUND que está impresso em vermelho na tecla "Z".

Introduza pois a seguinte linha:

```
65 SOUND 1,10
```

e execute (RUN) o programa novamente.

Com a introdução desta linha emite-se (SOUND) o tom de número 10 durante 1 segundo após cada impressão do seu nome.

Altere a linha 65 para

```
65 SOUND.3, coluna
```

O.I. - Mantenha abaixada a tecla **SYMBOL SHIFT** enquanto você aperta a tecla "M" para se obter o ponto decimal que deve aparecer antes do "3".

Execute (RUN) o programa novamente.

Hummmmm...

Interessante, mas não é exatamente o que se pretendia.

Tem-se uma "trilha" de letras iniciais que é um estorvo e além do mais surge a estranha mensagem de erro

```
B inteiro excede limite 50:1
```

Antes de iniciar a eliminação desta "mancada" seria interessante que você realmente compreendesse como funciona o *nono programa*.

A trilha de letras (sem nenhum intervalo) é provocada pelo fato de que ninguém ensinou e tão pouco pediu ao TK 90X para apagar o nome antigo antes de imprimir o mesmo em uma outra posição.

Desta forma execute (RUN) o programa mais uma vez porém quando o TK 90X lhe perguntar:

"Qual é o seu primeiro nome? "

pressiona inicialmente a tecla SPACE (espaço em branco) uma vez, antes que você entre com o seu nome digamos:

Sergio"

Execute (RUN) agora e não vá dizer que o seu TK 90X é um mágico.

O que ele é realmente é artificioso...

Como provavelmente você poderá esquecer deste espaço em branco (mágico) eu lhe indicarei no próximo programa exemplo, o de nº 10, uma forma do TK 90X lhe colocar este espaço..

Desta forma guarde (SAVE) este programa em uma fita cassete ou então deixe na memória do TK 90X pois no Capítulo 15 iremos utilizar o mesmo.

#### Detalhes sobre o nono programa

Linha 10 → Isto já é muito simples. REM lhe permite escrever algo que você acha que é explicativo ou indicativo para o programa como por exemplo o título do mesmo.

Qualquer coisa que for teclada após a palavra-chave REM é ignorada pelo TK 90X.

Linha 20 → É aí que se informou ao TK 90X em que linha da tela deve ser feita a impressão.

Escolhemos a linha de número 11 que na realidade é a 12ª linha (veja a Figura 2).

O número que colocamos na caixa de memória com o título "linha" é um número legal para a linha pois está no intervalo de 0 até 21.

Tente por conta própria mudar esta instrução dando a variável "linha" valores próximos de 0 (zero) ou de 21 para começar a ter um bom domínio da tela...

Linha 30 → Com o auxílio da instrução LET "ajusta-se" a posição inicial da variável "coluna" no ponto 0 (zero).

Foi escolhido este valor, para que o seu nome se movimente da esquerda para a direita ocupando toda a tela.

Linha 40 → Usou-se INPUT para parar o programa enquanto você tecla o seu nome.

O TK 90X guarda o seu nome na caixa de memória rotulada com n\$.

Desta forma, toda vez que você quiser que seja impresso o seu nome basta ordenar ao TK 90X para executar o comando PRINT n\$.

Linha 50 → Aí se indica ao TK 90X onde exatamente você quer a impressão do seu nome, que no caso particular terá a sua primeira impressão começando no "retângulo" definido pelo cruzamento da linha de número 11 com a coluna de número 0 (zero).

Desta forma PRINT AT linha, coluna terá inicialmente o significado PRINT AT 11,0.

Em outras palavras: "imprima (PRINT) na (AT) linha 11 a partir da coluna 0".

Como o valor da variável "coluna" irá aumentar de uma unidade em cada volta teremos comandos do tipo PRINT AT 11,1, PRINT AT 11,2, PRINT AT 11,3 etc. o que significa imprima na linha 11 a partir da coluna 1, imprima na linha 11 a partir da coluna 2, etc. e o seu nome começa a se deslocar.

Linha 60 → A instrução LET coluna = coluna + 1 não tem muito sentido em Álgebra, pois é capaz de você chegar a conclusão que  $1=0!?!?$

Na realidade não é nada disto.

O TK 90X entende esta instrução como: "em cada volta acrescenta ao conteúdo da caixa de memória com o rótulo "coluna" uma unidade ou ainda o novo valor de "coluna" é o antigo mais uma unidade".

O TK 90X irá assim em cada volta efetuando as seguintes somas: coluna é igual a nada (zero) mais 1 (um) na 1ª volta, um mais um na 2ª volta, dois mais um na 3ª volta, etc. até que o programa pare.

Você não esqueceu que o maior valor da variável coluna é igual a 31, não é?

Aliás,, se o nome-teste for "Sergio" introduza as seguintes instruções

```
55 IF Coluna < 25 THEN GOTO 60
```

```
56 PRINT AT linha, 25; "██████"
```

```
57 LET coluna = 0
```

```
58 GOTO 50
```

sete espaços em branco

que o seu programa não parará jamais a não ser que você queira isto!?!?!?

Linha 65 → Esta linha foi acrescentada para diminuir um pouco a velocidade com que é impresso o nome.

Enquanto o TK 90X está emitindo um som ele não pode mais fazer nada ao mesmo tempo.

Desta forma tudo para durante uma fração de segundo ou alguns segundos enquanto o TK 90X soa um "barulhinho".

Se o TK 90X está emitindo um som (SOUND) curto, obviamente que o atraso provocado no programa é



bem pequeno.

Entretanto se você mandar fazer um "SOOOOM" bem longo o atraso na execução do programa é bem sensível.

Os números após a instrução SOUND significam

SOUND duração, altura

onde "duração" significa quantos segundos você quer que dure a nota e "altura" significa o quão alto ou baixa é a nota que você quer ouvir.

Caso você queira uma nota bem curta, pode teclar

SOUND.3, coluna

que significa: "emita durante 0,3 segundos uma nota com um valor que você irá encontrar na caixa de memória com o rótulo "coluna".

Como o valor da variável "coluna" é aumentado de 1 unidade em cada volta que o TK 90X atinge a linha 60, o tom da nota aumenta de uma unidade cada vez que for emitida.

Só para sentir melhor e como uma experiência instrutiva modifique a linha 65 de tal forma que possa ter outros valores para o SOUND.

A duração pode variar de 0 até 10.

A altura (tom) pode ser qualquer número de -60 até +69.

Frações destes números também são aceitas pelo TK 90X.

Escreva por exemplo

$\left\{ \begin{array}{l} 65 \text{ SOUND } 3.5, 29.43 \\ \text{ou} \\ 65 \text{ SOUND } 5, -18 \\ \text{ou ainda} \\ 65 \text{ SOUND } 3, \text{coluna} + 3 \end{array} \right.$

Se você entende um pouco de música, caso o valor que você atribuir a altura for zero terá o "dó" do meio.

Portanto qualquer valor negativo (como -8) será uma nota mais baixa do que o "middle C" em um piano.

É claro também que se você escrever algo como SOUND 0,15 não terá muita utilidade pois a nota que dura 0 (zero) segundos é um pouco difícil de ouvir, não é?

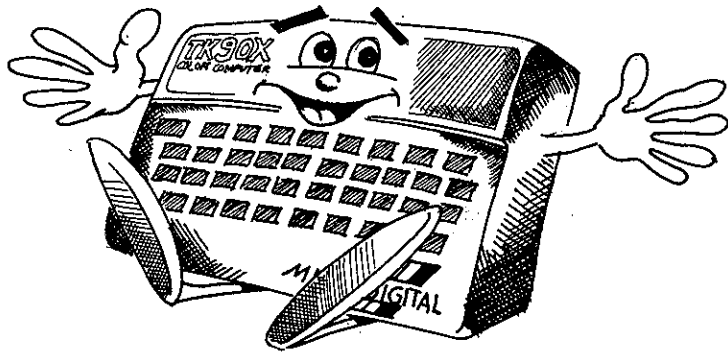
Linha 70 → O GOTO envia o TK 90X de volta a linha de número 50 na qual ordena-se para imprimir o seu nome outra vez, porém agora como o valor da "coluna" foi aumentado de 1 (linha 60) a impressão do nome sai um espaço a direita em relação a última posição. Não esqueça então que a grande ênfase neste capítulo foi para as instruções:

PRINT AT → Diz ao TK 90X para imprimir na linha e coluna especificados após o AT (o AT está em vermelho na tecla "I").

SOUND → Diz ao TK 90X para soar o alto falante em uma duração e altura (ou frequência) especificadas. O SOUND está abaixo da tecla Z (em vermelho).

## Capítulo 16

### Conectando e cortando STRINGS



O TK 90X pode juntar ou melhor concatenar duas ou mais cadeias de caracteres para desta forma criar frases.

Desta forma o TK 90X pode "somar" uma STRING contendo as letras "Ale" com uma outra STRING contendo "xandre" para formar o nome "Alexandre".

Pode-se inclusive ter uma STRING que dentro não tenham nem letras ou qualquer outro tipo de caractere, ou seja ter entre aspas um (ou mais) espaço(s) em branco.

Aliás isto é muito útil para que se tenha uma certa facilidade na leitura.

#### PROGRAMA 10

```

10 LET A$ = "Alexandre"
20 LET E$ = "Do grego : Alexandros"
30 LET S$ = "Significado:"
40 LET D$ = "defensor da espécie" : LET B$ = "□":
   LET T$ = "-"
50 PRINT "Aí vai uma explicação de um nome impor-
   tante, aliás o nome já foi a alma das pessoas!"
60 PRINT : PRINT
70 PRINT A$ + T$ + B$ + B$ + E$
80 PRINT S$ + B$ + D$

```

Execute e veja se sai isto:

Ai vai uma explicação de um nome importante, aliás o nome já foi a alma das pessoas.

Alexandre - Do grego: Alexandros  
Significado: defensor da especie



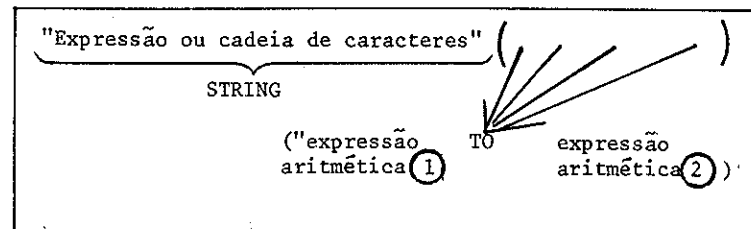
- 1) Nas linhas 40 e 60 aparecem dois pontos (":") que permitem que se obtenha o que se pode chamar de linha de instruções múltiplas ou seja em uma única linha lógica de programa termos várias instruções.
- 2) Nas linhas 70 e 80 foi feita a concatenação das STRINGS A\$, T\$, B\$ (espaço em branco) E\$ (linha 70) e S\$, B\$ e D\$ (linha 80).



É possível extrair uma subSTRING ou modificar uma STRING usando uma nova operação que podemos chamar de corte (em inglês "slicing") da STRING original.

Em outras palavras é possível determinar uma subSTRING a partir de uma STRING maior, aplicando a função TO (e por favor não confunda este TO com outras aplicações da função TO em outros comandos...).

A forma geral para o corte de uma STRING é



Assim se você escrever por exemplo

```
PRINT "123456789"(4 TO 7) (ENTER)
```

Sairá a subSTRING "4567" ou seja começando no quarto caractere e terminando no 7.

Por outro lado se você escrever:

```
"Victor" (1 TO 3) (ENTER)
```

será exibida a subSTRING "Vic" que representa os três primeiros caracteres da STRING "Victor".

O início e o fim de uma STRING são frequentemente utilizadas para formar subSTRINGS pelo corte e caso você não tenha especificado o início [expressão aritmética (1)] ou o fim [expressão aritmética (2)] o seu TK 90X pode assumir algumas coisas como está indicado abaixo:

```
"12345678" (TO 5) significa "12345678" (1 TO 5) e que dá "12345"
```

```
"12345678" (3 TO) significa "12345678" (3 TO 8) e que dá "345678"
```

```
"12345678" (TO) significa "12345678" (1 TO 8) e que dá "12345678"
```

"12345678" (7) significa "12345678" (7 TO 7) e que dá "7"

Se você caro(a) leitor(leitora) não percebeu ainda a "função corte" da STRING em toda a sua amplitude em primeiro lugar "bom dia!" ou seja acorde...

Acordou?

Aí vão mais algumas ilustrações:

10 LET F\$ = "Não se deve inventar" Resultado  
 20 PRINT F\$ (5 TO) → Exibe-se a subSTRING  
 "se deve inventar"

10 LET F\$ = "Não se deve inventar"  
 20 PRINT F\$ (6 TO 14) → Exibe-se a subSTRING  
 "e deve in"

10 LET F\$ = "Não se deve inventar"  
 20 PRINT F\$ (7 TO) → Exibe-se a subSTRING  
 "deve inventar"

Está clareando, não é estimado(a) leitor (leitora)?

Tudo se tornará óbvio (acho eu...) após o seguinte programa-exemplo.

### PROGRAMA 11

Vamos usar a função corte para se imprimir o nome de um mês dado o seu número (isto é para setembro (Set) você deve entrar com o número 9).

O resultado exibido será do tipo

mês 9 é Set

Você irá ouvir um rápido efeito sonoro para cada saída apresentada na tela.

Aí vai o programinha:

```
10 LET n$ = "JanFevMarAbrMaiJunJulAgoSetOutNovDez"
20 INPUT "Entre com o número do mês "; mês
30 PRINT "mês "; mês; " é "; n$(1+3*(mês-1) TO
  3*mês)
40 FOR j = 69 TO - 60 STEP -1
50 SOUND .1, j
60 NEXT J
70 GOTO 20
```

### Algumas explicações sobre o programa 11

- 1) A amplitude numérica para a variável mês é de 1 até 12 (números inteiros) para se obter a abreviatura correta para o mês em questão. A forma como opera a linha 30 é a seguinte: se você entrar com mês de número 5 (maio) então  $1+3*(mês-1) = 1 + 3*(5-1) = 13$  e  $3*mês=15$  resultando daí um corte na STRING entre as posições 13 e 15 ou seja é exatamente onde está a abreviatura "Mai" isto é a subSTRING exibida.
- 2) É no Capítulo 17 que você aprenderá como funcionam as linhas 40 e 60 (FOR-NEXT). Por enquanto não esqueça como funciona o SOUND caso tenha esquecido volte ao Capítulo 15.



# Capítulo 17

## Preparando-se agora para (FOR) o seu próximo (NEXT) grande truque no TK 90X

Em computação (como aliás em tudo...) existe sempre uma forma melhor de se conseguir fazer uma coisa.

Neste capítulo você irá conhecer uma "hera" (do latim hera que significa "senhora") instrução que irá capacitá-lo(la) a mover coisas para baixo e para cima ou para esquerda e para direita na tela, deslocar as suas naves e as suas armas para qualquer ponto da tela, etc.

Estou me referindo a instrução

FOR-TO/NEXT

que talvez seja uma das mais poderosas do BASIC.

A instrução FOR/NEXT é constituída de dois comandos: o comando FOR e o comando NEXT.

A instrução FOR-TO/NEXT é aparentemente complicada porém é de grande importância para algumas aplicações e de saída você irá teclar e executar o programa 12.

PROGRAMA 12

1Ø LET n = 1

2Ø PRINT "Eu não queria ter xará!"

```

30 LET n = n+1
40 IF n = 14 THEN GOTO 60
50 GOTO 20
60 REM *** o programa vai descansar um pouco**

```

Até aqui você já conhece tudo e ao executar o programa 12 como está escrito você verá na tela repetido treze vezes a mensagem

"Eu não queria ter xará!"

Vamos executar agora o mesmo programa usando o FOR-TO/NEXT.

O seu aspecto será agora:

```

10 FOR n = 1 TO 13
20 PRINT "Eu não queria ter xará!"
30 NEXT n
40 REM *** O programa vai descansar um pouco"

```

Acho que você já notou deste pequeno exemplo que o FOR-TO/NEXT lhe economizou, para começar, duas linhas completas do programa.

Porém ele faz muito mais do que isto.  
Entenda agora como funciona o programa.

- 1) Na linha 10 o TK 90X atribui à variável n inicialmente o valor 1 e passa para a próxima linha.
- 2) Na linha 20 o programa encontra a instrução PRINT e imprime a mensagem "Eu não queria ter xará!", e passa para a linha seguinte.
- 3) Na linha 30 encontra-se o comando NEXT ("next" significa próximo em inglês) que faz com que o novo valor de n seja n+1 comparando o mesmo com o valor final que no nosso caso particular é 13.

Enquanto este valor de n for menor ou igual que 13 o programa volta automaticamente para a linha 10.

Na 1ª volta ao se chegar à linha 30 o valor de n é 1 e sai daí com o valor n=2 que é menor que 13. Com isto repete-se o ciclo e a mensagem "Eu não queria ter xará!" é impressa pela 2ª vez.

- 4) O ciclo se repete até que o valor (final) de n se torne 13 quando então ao chegar na linha 30 ele se torna 14 e ao voltar para a linha 10 ele é maior do que o limite superior especificado fazendo com que o TK 90X abandone o "LOOP" (laço ou malha) prosseguindo o programa na primeira instrução após o NEXT.



- 1) As vezes, interessa que a cadência com que uma variável dentro da instrução FOR vá tomando valores que vão aumentando com incremento diferente da unidade. Para esta finalidade existe à sua disposição a instrução STEP que permite que a variável seja incrementada (aumentada) ou decrementada (diminuída) à sua vontade.

Assim se você escrever:

```

[ FOR x = 20 TO 40 STEP 5
  :
  NEXT x

```

o seu TK 90X entenderá que o programa terá neste trecho cinco repetições, em cada uma delas a variável x toma os valores 20, 25, 30, 35 e 40 respectivamente e tudo isto devido à instrução STEP que depois de si tem o incremento 5.

Pode-se também diminuir o valor de uma variável em cada volta e para tanto basta que o número após o STEP seja negativo.

Assim se você escrever:

```
FOR x = 100 TO 50 STEP -10
:
NEXT x
```

o seu TK 90X entenderá que o programa terá neste trecho seis repetições, em cada uma delas a variável x tomará os valores 100, 90, 80, 70, 60, 50 respectivamente e como se pode perceber em cada volta a variável tem o seu valor diminuído (decrementado) de 10 até atingir o valor limite 50.

- 2) Dentro de uma malha FOR-TO/NEXT pode-se embutir ou aninhar um ou mais "loops" FOR-TO/NEXT com a condição fundamental que os mesmos não se cruzem (veja a Figura 4).

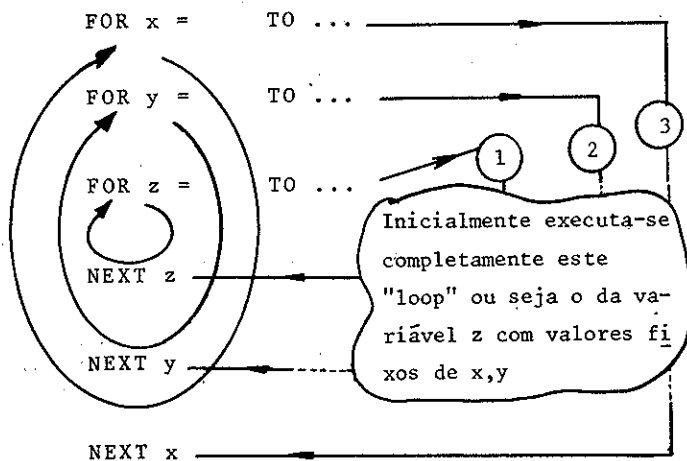


Figura 4 - São dadas todas as voltas em 1 para se dar uma volta seguinte em 2 e depois que se dão todas as voltas completas em 2 se dá uma nova volta em 3. E assim seguirá até que tudo seja completado...

- 3) Bem, agora dá para entender as linhas 40, 50 e 60 do programa 11.

A variável j tem valor inicial 69 e vai diminuindo em cada volta este valor de 1 até atingir o valor -60.

Com isto cada vez que se passa pela linha 50 gera-se som através do alto falante do seu televisor e isto devido a instrução 50 SOUND .1,j onde 0.1 dá a duração da emissão do som (em segundos) e j é a frequência da nota.

### PROGRAMA 13

Neste programa no qual se usam dois "loops" um aninhado dentro do outro vão ser calculados os 70 primeiros números primos de 2 em diante.

Como a lógica do mesmo é um pouco complexa desenhamos ao lado dos números de linha os possíveis desvios do fluxo do programa.

Acho que com isto, como a explicação dada há pouco e a que surgirá após o 13º programa você atencioso(a) leitor(a) deglutirá mais um...

```
10 LET p = 1
20 FOR v=1 TO 70
30 LET p = p+2
40 FOR f=3 TO SQR p STEP 2
50 IF p=f * INT (p/f) THEN GOTO 30
60 NEXT f
70 PRINT TAB 8*v; p;
80 NEXT v
```

Com esta função calcula-se a raiz quadrada de um número

Explicação complementar

O "loop" principal do programa é o que vai das linhas 20 até 80 e conta apenas o número de primos que devem ser encontrados sendo controlado pela variável contadora "v".

Dentro do "loop" interno, a variável "p" é o número que está sendo testado para verificar se é primo (você sabe o que significa um número primo?).

Começou-se com um valor ímpar (3) e incrementou-se o mesmo em cada volta de dois (2) para evitar no programa o teste de números pares que obviamente *não são primos*.

Para testar e saber se p é primo necessitou-se do "loop" de 40 a 60 no qual se verifica se qualquer número inteiro no intervalo entre 3 e a raiz quadrada de p é um fator de p.

Se isto ocorrer, p não é primo e o programa salta (na realidade volta) para a linha 30 para se ter o próximo valor de p que deve ser testado, o qual vem aumentado de 2 em relação ao último.

Note que o BASIC do TK 90X permite que se possa sair de uma malha FOR-TO/NEXT sem ter terminado o mesmo.

Isto é as vezes bem difícil, para não dizer impossível em algumas versões de BASIC usadas por certas máquinas...

Caso não sejam encontrados fatores então p é primo e o seu valor é impresso na linha 70.

Vale a pena neste momento falar um pouco sobre a instrução TAB.

Ela permite ao seu TK 90X iniciar a impressão em qualquer posição de uma dada linha.

Em geral TAB n desloca a posição de impressão para a frente ao longo da linha e a partir da coluna n.

As colunas (como você sabe) são numeradas de 0 até 31 e isto da esquerda para a direita.

Caso a posição de impressão passe a coluna de número 31, por exemplo tendo-se um TAB 43) isto significa que n passa da 1ª linha para a segunda e contam-se as restantes posições para começar a impressão.

Neste caso particular, para TAB 43 se começaria a impressão na 2ª linha a partir da coluna 11.

Para sentir melhor o que ocorre teclando e execute o seguinte mini-programinha.

**PROGRAMA 14**

```
10 PRINT TAB 0; "ALE";
20 PRINT TAB 32; "ALE";
30 PRINT TAB 43; "ALE";
40 PRINT TAB 75; "ALE";
50 PRINT TAB 192; "ALE";
```

Conclusão

No caso de que se tenha um valor maior que o permitido os dados se deslocam para a linha seguinte sendo o referencial a última linha ocupada.

Ao executar você deve ter a saída indicada na Figura 5.



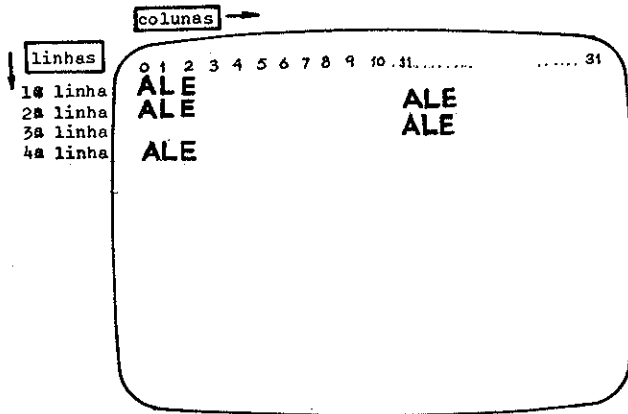


Figura 5

Conclusão

Portanto o FOR-TO/NEXT diz ao TK 90X para executar o "loop" (laço) entre o FOR e o NEXT usando valores especificados.

Note que o FOR está na **tecla F**, o NEXT está na **tecla N**, o STEP está na **tecla D** e o TO está também na **tecla F** e em vermelho.

Vá agora ao Capítulo 23 e tecle os programas 5 e 6 para dominar mais ainda o FOR-TO/NEXT.

Convém fazer isto antes de passar ao Capítulo 18.



## Capítulo 18

### Movendo-se na tela com o auxílio do FOR-TO/NEXT

Agora que você já sabe como se trabalha com os "loops" FOR-TO/NEXT vamos tentar com o auxílio dos mesmos mover o seu nome na tela para logo depois chegar ao seu primeiro joguinho real.

Não é o que você está esperando?

Wonderfuuuuuul!!!

Para início de conversa não esqueça que um ciclo ("loop") sem corpo é um ciclo de atraso.

Mas ele pode, mesmo assim ser útil tornando a execução de um programa mais lento.

É o caso de:

```

10
:
100 PRINT "Início"
110 FOR Z=1 TO 999
120 NEXT Z
:
:
200 PRINT "Fim"

```

onde com as linhas 110 até 120 provoca-se uma *pausa*.

No programa 9, quando se mostrou entre outras coisas como se podia deslocar o seu nome no TK 90X teve que se acrescentar um espaço em branco

antes do nome.

Foi um truque "original", sem dúvida nenhuma...

O movimento foi da esquerda para a direita e agora se quiser ir no outro sentido um espaço em branco deverá ser acrescentado após o nome, caso contrário você terá uma linha de letras indesejáveis.

Vamos repetir o *programa 9* usando agora o FOR-TO/NEXT.

Tecla então:

```
10 REM * Fazendo o seu nome andar da direita para
    a esquerda*
20 INPUT "Qual é c seu primeiro nome?"; n$
30 LET n$ = n$ + "□"
```

Neste ponto você pode usar o laço FOR-TO/NEXT para contar do fim para o começo:

```
40 FOR c = 22 TO 0 STEP -1
```

Você lembra que para contar do fim para o começo é preciso colocar inicialmente o número maior (22 neste caso) e depois o número menor (neste caso o 0) após a instrução TO.

Depois disto você precisa ainda dizer ao TK 90X em quantos passos ou etapas deseja que seja feita a passagem do número maior para o menor.

Como agora é do fim (do maior) para o começo (para o menor) o passo terá um valor negativo e o escolhido foi -1.

Pode também informar ao TK 90X onde você quer imprimir o seu nome.

Digamos na linha de número 7 ou seja na oitava linha e a partir da coluna c (o valor inicial de c é 22 e supõe-se que o seu primeiro nome não tem mais que 10 caracteres).

Acrescente a instrução

```
50 PRINT AT 7,c; n$
```

Não é nada ruim ter um pouco de som para tornar o programa um pouco mais vagaroso e é por isto que se tem a linha 60.

```
60 SOUND .5,c : SOUND .125, c :
    SOUND .016,c
```

Você está apto a fechar o ciclo("loop") com a linha:

```
70 NEXT c
```

Você tem alguma idéia porque usou-se a variável de controle "c" do "loop" tanto na linha 50 como na linha 60?

Resp.: Em primeiro lugar se nós tivéssemos o número de linha *l* e o mesmo fosse importante seria usado...

O que temos e precisamos é variar a coluna (c) e por isto aproveitamos esta variável como controladora para o "loop", para a coluna no PRINT AT e ainda para a frequência dentro do SOUND aplicando a lei do mínimo esforço e máxima utilidade...

Resta agora fazer o nome voltar da esquerda para a direita e ficar assim num "zig-zag" eterno...

Para tanto acrescente as seguintes linhas

```

80 LET n$ = "□" + n$
90 FOR c = 0 TO 22
100 PRINT AT 7,c; n$
105 SOUND .5, c : SOUND .125, c : SOUND .016, c
110 NEXT c
120 FOR Z=1 TO 999 : NEXT Z
130 CLS
140 GOTO 30

```

Um ciclo sem corpo  
ou seja uma pequena  
pausa

Se você quiser começar com outro  
nome faça aqui GOTO 20

Caso queira interromper este programa não use **SYMBOL SHIFT** e **A** pois não irá funcionar este "breque".

Aqui você deve usar **CAPS SHIFT** e **SPACE** para interromper o "loop".

Vamos ver agora como se pode mexer para cima e para baixo.

Antes de mais nada lembre que você é um(a) "personagem" que não vive só o presente e pensa muito no futuro pois desta forma, sem dúvida, será recompensado(a) por sua diversidade e principalmente pela originalidade em algumas atitudes.

Deixando a filosofia de lado, se você quiser mover uma palavra ou uma mensagem para cima ou para baixo na tela da T.V. precisa imprimir uma linha de espaços em branco em cima da palavra ou da mensagem "antiga" para apagá-la antes que a mesma seja impressa em uma nova posição.

No seu TK 90X o conjunto de muitos espaços em branco ou seja um grande vazio vale muito e é bem diferente do nada ou seja onde não há vazio nenhum...

Nomes as vezes podem ser muito estranhos e o deputado federal Áureo de Melo é o autor do proje-

to de lei (1965) que quer proibir o registro de nomes ridículos, humilhantes, espalhafatosos, eróticos ou proibidos.

Segundo o projeto, quem tentar registrar um(a) filho(a) com os nomes proibidos terá que pagar multa e não será apenas o pai mas também o escrivão que fizer o registro e o juiz que aprovar o dito nome.

Entre alguns dos nomes estranhos, que servirão de amostra para o programa 15, analise estes:

Amim Amon Amado  
Nascente Nascido Puro  
Lança Perfume Rodometálico de Andrade  
Um Dois Três de Oliveira Quatro  
José Casou de Calças Curtas  
Rolando Emídio da Torre da Igreja etc.

#### PROGRAMA 15

```

10 REM***fazendo o nome subir***
20 INPUT "Entre com o nome□";n$
30 FOR L=21 TO 0 STEP -1
40 PRINT AT L,7; n$
50 SOUND .2,L: SOUND .1,L
60 PRINT AT L,7; "□□□□□□□□□□□□□□□□ "
70 NEXT L
80 FOR Z=1 TO 199 : NEXT Z
85 REM***fazendo o nome descer***
90 FOR L=1 TO 21
100 PRINT AT L,7; n$
110 SOUND .2,L: SOUND .1,L : SOUND .125,L
115 PRINT AT L,7; "□□□□□□□□□□□□□□□□ "
120 NEXT L
130 FOR Z=1 TO 199 : NEXT Z
140 GOTO 20

```

Por exemplo  
Amim Amon Amado

Execute e compreenda o movimento na vertical para estar preparado para o seu primeiro jogo (aliás o que tem de jogo no livro dos meus filhos Sérgio e Victor não é brincadeira xará...).

O cuidado que você deve tomar neste programa ao entrar com o nome (n\$) e ver se ele não tem mais do que 15 caracteres quando então deve-se aumentar os espaços em branco da linha 60.



Não, não, não!!!

Não trema, eu sei que está chegando a hora de você começar a se capacitar totalmente para poder escrever os seus próprios programas de jogos.

Mais algumas instruções e você estará "maduro"(a) para este fim...

### PROGRAMA 16

```

10 REM***ARMA***
20 INPUT "Entre com uma palavra comprida";p$
30 PRINT AT 0,12; p$
40 LET arma = 16
50 PRINT AT 21, arma; "□↑□"
60 PAUSE 0
70 LET m$ = INKEY$

```

por exemplo  
Guaratinguetã

```

80 IF m$ = "5" THEN LET arma=arma -1 : GOTO 50
90 IF m$ = "8" THEN LET arma=arma +1 : GOTO 50
100 IF m$ <>"0" THEN GOTO 50
110 FOR L = 20 to 0 STEP -2
120 PRINT AT L, arma+1; "!"
130 SOUND .05,L
140 PRINT AT L, arma +1; "□"
150 NEXT L
160 GOTO 60

```

Antes de executar este programa veja se você bateu dois pontos ":" e não ponto e vírgula";"no final das linhas 80 e 90.

Como já comentei anteriormente o uso de dois pontos ":" lhe permite incluir mais do que uma instrução na mesma linha.

Você ganha um abraço (ou um beijo, isto depende do sexo...) se responder porque houve necessidade de colocar mais que uma instrução nestas duas linhas!

Apareceram duas instruções novas no programa 16.

A primeira é **PAUSE** que é uma palavra-chave que está na tecla M.

A forma geral desta instrução é **PAUSE t** onde t é tempo em  $\frac{1}{50}$  de segundo que você quer que o TK 90X espere.

Então como o próprio nome diz a instrução PAUSE gera uma parada ou pausa determinada pelo valor do argumento.

PAUSE 0 equivale a um tempo indefinido de espera e só pode ser interrompido pelo acionamento de alguma tecla (com exceção da SHIFT).

A instrução INKEY\$ ou como alguns costumam chamar a variável INKEY\$ ou ainda a função INKEY\$ faz com que seja executada a leitura do teclado obtendo uma STRING com um único caractere como resultado.

Se o teclado não foi acionado ter-se-á como resultado uma STRING totalmente vazia ou seja sem nenhum espaço em branco.

A função INKEY\$ opera de forma um pouco semelhante a instrução INPUT com uma diferença fundamental que ela entra em ação assim que a primeira tecla for pressionada e não necessita que se aperte `ENTER` para que isto ocorra.

Qualquer informação indo para o TK 90X usando-se INKEY\$ é tratada como uma STRING, independente do fato da tecla pressionada ser de uma letra ou de um número.

Desta forma, é preciso que ela seja armazenada na caixa de memória do tipo STRING, representada com apenas uma letra seguida do sinal \$.

Execute agora o programa e distraia-se com o mesmo.

Use a tecla `5` para mover a sua arma `↑` para a esquerda e a tecla `8` para mover a arma para a direita e a tecla `0` para disparar o míssil.

Se você olhar para a linha 80 vai perceber logo que toda vez que a tecla `5` é apertada, o número na caixa de memória com o rótulo "arma" terá subtraído de si a quantidade 1.

Como o seu valor inicial é 16 cada vez que se aperta a tecla `5` ter-se-á:

```
arma = arma - 1
```

e assim na primeira vez, "arma" terá valor 15 e se você continuar apertando a tecla `5` a variável "arma" poderá tomar os valores 14,13,12,11,...0.

Caso você continuar apertando o valor atribuído será -1, -2, -3, etc. e que estão todos fora da tela!?!?!.

Não faça isto pois entre outras coisas não há em que atirar...

O mesmo pode ocorrer se você apertar muito a tecla `8` só que nesse caso a variável "arma" a cada pressionada da tecla `8` é aumentada de uma unidade porém você não deve ter para "arma" valor maior que 31 pois neste caso estará fora da tela, a direita da mesma.

Para evitar que isto aconteça, você deve fazer um teste para ver se a sua arma não está no fim da linha.

Em outras palavras, você só quer mover a arma se (IF) a tecla 8 for pressionada e (AND) a arma não está no fim (extrema direita).

Dessa forma usando o símbolo vermelho AND na tecla Y mude a linha 80 para

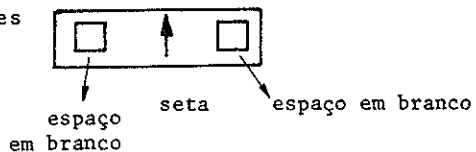
```
80 IF m$ = "5" AND arma > 0 THEN LET arma=arma - 1
: GOTO 50
```

Com esta instrução a arma move-se somente se ambos m\$ = "5" e (AND) a posição da arma é maior do que 0.

O mesmo deve ser feito para a linha 90.

Neste instante você terá que escolher um número para o seu limite com muito cuidado, pois você precisa levar em conta o comprimento da "coisa" que está imprimindo.

Uma forma de estar seguro é não ultrapassar a coluna 31 e como a arma foi construída com três caracteres



o limite máximo passa a ser a coluna 29 que não deve portanto *ser ultrapassada*.

A linha 90 deve ser:

```
90 IF m$ = "8" AND arma < 29 THEN LET arma=arma+1
: GOTO 50
```

Agora você pode executar (RUN) o programa e treinar a *sua pontaria*.

É claro que o programa está muito estático. Cabe a você inteligente leitor(a) juntar o que foi ensinado no programa 9 ou seja como fazer o nome se mover da esquerda para direita e vice-versa que aí então o dinamismo fará com que você erre alguns tiros e o jogo se tornará mais interessante...

Você pode também no lugar de uma palavra construir uma STRING com caracteres "<", ">", "=" dando assim um aspecto de uma nave espacial.

Isto também melhorará as coisas.

Mãos a obra o(a) original é você !!

#### NOVAS INSTRUÇÕES (Resumo)

**PAUSE t** → Diz ao TK 90X para esperar durante um tempo fixo ou até que uma tecla seja pressionada (aquela que lhe pa-

reçar mais simpática, menos a SHIFT...) O número que segue PAUSE especifica quantos  $\frac{1}{50}$  do segundo deve-se esperar. PAUSE está na **tecla M**.

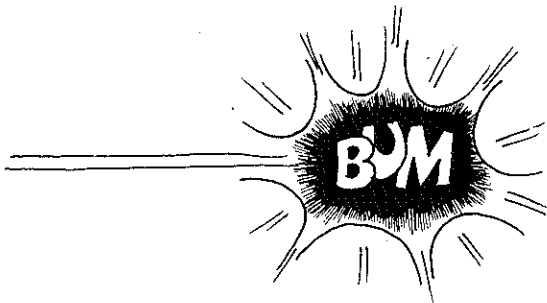
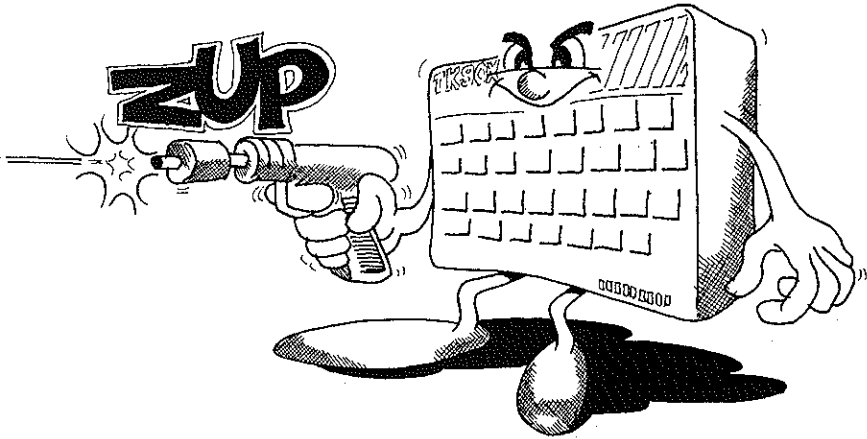
**INKEY\$** → Diz ao TK 90X para tomar a próxima tecla pressionada como uma entrada STRING e não esperar pelo ENTER. Está acima da **tecla N**.

**AND** → É usada junto com o IF-THEN para fazer com que o TK 90X só faça alguma coisa somente se ambas as instruções forem verdadeiras. Está na **tecla Y** AND e um operador lógico.



## Capítulo 19

### Usando caracteres gráficos que permitam simular uma explosão



Vamos transformar aquela palavra (ou palavra) que você mandou no programa 16 em uma "caricatura" parecida com um avião.

Para isto precisaremos do modo gráfico.

Note que acima da **tecla 9** está a palavra GRAPHICS impressa em branco.

Mantenha abaixada a tecla **CAPS SHIFT** enquanto aperta a **tecla 9**.

Você notou que o cursor mudou para um piscante **G**?

Notou, mesmo!!!!!!

É isto que lhe indicará no TK 90X que você está no modo gráfico.

Em cada tecla de 1 até 8 existe um pequeno símbolo branco como o que está indicado na Figura 6.



Figura 6

Aperte a **tecla 1** e observe o pequeno quadrado que aparece.

Agora aperte **CAPS SHIFT** e **1** juntas.

Desta vez, o pequeno quadrado não é impresso, porém o resto do caractere é.

Caso o TK 90X tenha um cursor piscante **G**, então ao se apertar uma tecla numérica ter-se-á, na cor do INK, o quadrado gráfico mostrado na cor escura naquela tecla.

Porém se você mantiver **CAPS SHIFT** abaixada, enquanto aperta uma tecla numérica, o TK 90X imprimirá na cor da instrução INK, aquilo que está impresso em branco naquela tecla.

Brinque um pouco com as teclas no modo gráfico (GRAPHICS) até que você esteja seguro(a) de que já compreendeu o que está acontecendo.

Enquanto estiver no modo **G**, o **0** ainda apaga (DELETE) e a tecla **9** faz voltar ao cursor normal ou seja desativar o modo gráfico.

### PROGRAMA 17

Realmente agora as coisas vão ficar um pouco mais emocionantes pois você verá um avião se movimentando na horizontal e poderá tentar destruí-lo simulando inclusive uma explosão.

Neste programa a função INKEY\$ terá um desempenho muito importante e a parte mais difícil da lógica está indicada no fluxograma da Figura 7.

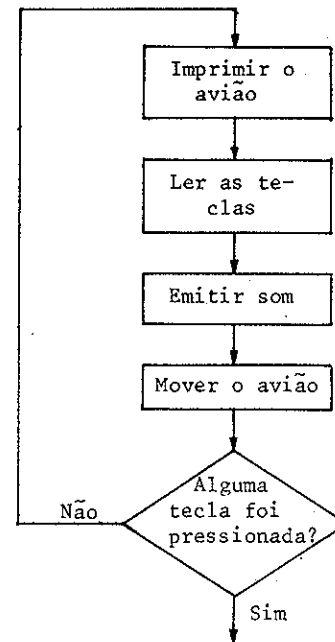


Figura 7 - As linhas 60 a 100 fazem o avião andar...

Aí vai o programa:

```

10 REM ** Explodindo um avião **
20 LET a$ = "  █ █ █ █ "
30 LET avião = 0
40 LET arma = 16
50 PRINT AT 21,arma; "  +  "
60 PRINT AT 2,avião; a$
65 LET m$ = INKEY$
70 SOUND .1, -20
80 LET avião = avião + 1
90 IF avião > 26 THEN PRINT AT 2,27; "████":
   LET avião = 0
100 IF m$ = "" THEN GOTO 60
110 IF m$ = "5" THEN LET arma=arma - 1 : GOTO 50
120 IF m$ = "8" THEN LET arma=arma + 1 : GOTO 50
130 IF m$ < > "0" THEN GOTO 50
  
```

estes caracteres são obtidos com  
 SPACE + CAPS SHIFT 1 → █  
 CAPS SHIFT 3 → █



```

140 FOR L = 20 TO 0 STEP -2
150 PRINT AT 2, avião; a$
155 IF SCREEN$ ( L, arma +1) < > "□" THEN GOTO 220
160 PRINT AT L, arma +1; "!"
170 SOUND .05,L : SOUND .05,-10
180 LET avião = avião +1
190 IF avião > 26 THEN PRINT AT 2,27; "□□□□":
    LET avião = 0
200 PRINT AT L, arma +1; "□"
210 NEXT L
215 GOTO 270
220 REM**aquí indica-se a explosão**
230 PRINT AT 2, avião; FLASH1; "■ ■ ■ ■"
240 FOR Z=1 TO 20
250 SOUND .01, RND*(-20)
260 NEXT Z
270 CLS
280 GOTO 20

```



Agora, execute o programa 17 e tente acertar no avião.

### Explicações adicionais

1) Você já notou, que no lugar da PAUSE que tinha na linha 60 no programa 16 foi colocada uma instrução PRINT AT, não é?

O fato é que uma pausa (PAUSE) é muito boa quando não há nada acontecendo enquanto o TK 90X está esperando por você.

Porém se você quiser que o avião se mova, deve ser um método um pouco diferente.

Uma forma de obter isto é colocando a função INKEY\$ em um laço.

Este "loop" será capaz de fazer duas ou três coisas em um espaço de tempo muito curto e pode também agir como uma parada até que uma tecla seja pressionada.

A maneira mais simples de entender o que INKEY\$ provoca é voltando a Figura 5 e acompanhando o programa.

É na linha 60 que começa este "loop" ("Imprimir o avião").

Neste laço utiliza-se INKEY\$ para testar se uma tecla foi pressionada.

Caso não tenha sido pressionada nenhuma tecla ouve-se o ruído do avião e o mesmo vai se deslocando.

Mas como é que o TK 90X sabe se você apertou alguma coisa?

Resp.: INKEY\$ dá uma STRING vazia se nenhuma tecla foi pressionada.

Uma STRING "nada" é formada apenas de dois conjuntos de aspas, um imediatamente após o outro SEM NENHUM ESPAÇO EM BRANCO NO MEIO ou seja:

```
"  "
```

São as linhas 80 e 90 que fazem o avião se deslocar.

2) Quando você executar (RUN) o programa você verá o avião se deslocando da esquerda para a direita (no céu...) até que você aperte a tecla 0.

Ao se pressionar esta tecla (0) saltar-se-á fora do laço que vai das linhas 50 até 130 e entra-se no "loop" do FOR-TO/NEXT que faz com que se desloque o míssil.

Entretanto, não é interessante que o avião fique "paralisado" enquanto o seu projétil se desloca para destruí-lo.

É por isto que existem as linhas 180 e 190 que fazem com que o avião se movimente, fugindo ou pelo menos não esperando a destruição *estático*.

3) Vamos agora a explosão...

Tudo que você deve fazer para isto é perguntar ao TK 90X para que o mesmo verifique se onde ele imprime a posição do projétil há algo diferente de espaço em branco.

Caso haja, só pode ser uma parte do avião e nestas condições deve-se desviar o programa para aquelas linhas que provocam a explosão (220 a 260).

A instrução que permite verificar se você acertou no avião é `SCREEN$`, a qual está numa palavra vermelha abaixo da `tecla K`.

A instrução `SCREEN$` utiliza os *mesmos números de linha e coluna* que são usados na instrução `PRINT AT`, porém eles devem aparecer entre *parênteses*.

A instrução `SCREEN$` diz ao TK 90X que tipo de caractere ele identifica em alguma posição da tela ou seja na posição da linha e coluna indicadas entre parênteses.

Caso não haja nada, ela diz ao TK 90X que aí está um espaço em branco, que aliás é o que existe na maior parte da tela...

Se por outro lado algo for detectado, sem dúvida é o avião.

Esta é a razão da existência da instrução de número de linha 155 que nos permite desviar o programa para a linha 220 onde está a rotina para a

explosão (linhas 220 até 260).

Para se obter os símbolos gráficos você deve entrar no modo gráfico e aí então apertar as seguintes teclas.

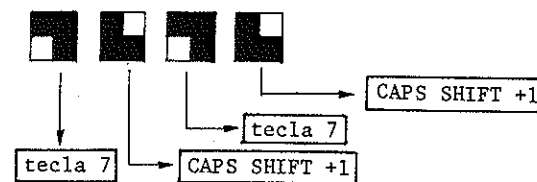


Figura 8

Acho que você querará jogar mais que uma vez e é por isto que limpei a tela (linha 270) e mandei começar tudo de novo.

Caso você queira um panorama mais colorido acrescente ao seu programa a linha 15.

```
15 BORDER 3 : INK 0 : PAPER 5
```

Uma versão bem mais sofisticada deste programa aparece no livro dos meus filhos Sergio e Victor "Jogos e Desenhos no TK 90X" - vol. 1.

4) O `FLASH` que aparece na linha 230 é um artifício utilizado para realçar um caractere, um conjunto de caracteres ou toda a tela sendo que

```
FLASH 1 significa ativado e FLASH 0 significa desativado
```

As novas instruções ou comandos neste capítulo foram:

**GRAPHICS** → Diz ao TK 90X para entrar ou sair do modo gráfico e é obtido com o

**CAPS SHIFT +9**

**FLASH** → Diz ao TK 90X para cintilar (FLASH) o ítem impresso a mesma taxa que a do cursor.

O FLASH está abaixo da **tecla V**

**SCREEN\$** → Pergunta ao TK 90X o que existe na tela em uma posição especificada.

Está abaixo da **tecla K**.

Vá agora ao Capítulo 23 e tecle o programa 7.



## Capítulo 20 Desenhando com linhas bem finas

A "forma plana" que foi desenhada no programa 17 era constituída de "blocos" bastante grandes e não parecia muito com um avião real.

Desenhar figuras com os caracteres gráficos é mais ou menos a mesma coisa que desenhar algo com lápis de desenho ("crayon") não apontado e bem grosso...

Para que as suas representações gráficas se tornem mais realistas você necessita de um lápis bem mais "fino" e com uma ponta bem apontada.

É agora que vou lhe ensinar como o TK 90X desenha figuras no modo de alta resolução ou seja com contornos representados por linhas bem delgadas.

O TK 90X é realmente "fora de série" visto que com ele pode-se facilmente misturar gráficos de alta resolução com o texto (letras, números e outros caracteres não gráficos).

As instruções de alta resolução para o TK 90X subdividem a tela em pontos ou *pixels*.

Imagine que cada caractere gráfico anterior seja subdividido em "quadrinhos" menores como ocorre em um tabuleiro de xadrez ou seja cada quadrinho tem tamanho  $\frac{1}{64}$  do caractere gráfico.

Para começar note que na **tecla Q** tem-se em branco a palavra-chave PLOT.

Esta instrução informa ao TK 90X para colocar um ponto em um lugar especificado da tela.

Para se especificar este ponto após a instrução PLOT você deve indicar a abscissa (x) e a ordenada (y) conforme se mostra na Figura 9.

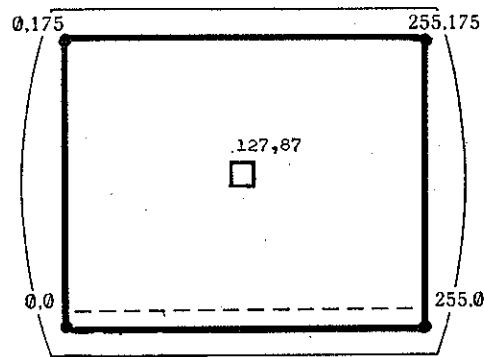


Figura 9

Portanto a instrução PLOT x,y diz ao TK 90X para preencher o pixel em x,y (você sabe, que costuma-se chamar a estes números de coordenadas do ponto...).

Tecle PLOT 0,0 **(ENTER)**

Você consegue ver um pequeno ponto no canto baixo da esquerda?

Uuuotimo!!!

Tecle agora PLOT 127,87 **(ENTER)**

Você tem agora um pequeno ponto no centro da tela.

Ao contrário dos números de linha e coluna para as instruções PRINT AT e SCREEN\$, que foram usadas em capítulos anteriores, as coordenadas da PLOT *não* tem a origem (0,0) no canto alto da esquerda.

Os números da tela no modo gráfico de alta resolução começam no canto baixo da esquerda, aliás como você esperto(a) leitor(a) já percebeu...

Os números na horizontal podem ir de 0 até 255.

(valores de x = abscissa) e os na vertical de 0 até 175.

(valores de y = ordenada).

Você verá a partir de agora a grande diferença.

### PROGRAMA 18

Este programa vai lhe permitir desenhar coisas na tela da sua T.V.

Muito mais do que isto, para que você não se sinta perdido(a), o TK 90X lhe mostrará sempre um *pixel* cintilando.

Use as teclas com as setas (5,6,7 e 8) para mover o cursor.

A medida que ele se desloca deixará um rastro, que pode ser um lindo desenho...

```
10 REM*** rascunho de um desenho***
20 BORDER 6 : CLS
30 LET x = 127 : LET y = 87
40 PLOT INVERSE 1 ; x,y
50 LET m$ = INKEY$
```

```

60 PLOT INVERSE 0; x,y
70 PRINT AT 5,10; "x=""; x; "y = "; y; ""
80 IF m$ = "" THEN GOTO 40
90 IF m$ = "5" AND x > 0 THEN LET x = x - 1
100 IF m$ = "8" AND x < 255 THEN LET x = x + 1
110 IF m$ = "7" AND y < 175 THEN LET y = y + 1
120 IF m$ = "6" AND y > 0 THEN LET y = y - 1
130 SOUND .25,0:SOUND.25,2:SOUND.025,5:SOUND.25,7
140 GOTO 40

```

Linda instrução musical

Execute este programa e ache o seu rumo o mais breve possível na tela...

### 0.1.

1) A instrução PLOT INVERSE 1; x,y diz ao TK 90X para inverter a cor dos caracteres com a cor do fundo quando o código após a INVERSE é 1.

Se depois do INVERSE aparecer o código 0 de sativa-se a inversão.

2) Se você quiser que os seus pontos tenham cor, pode usar uma instrução do tipo:

PLOT INK código da cor; x,y

este valor pode ser de 0 a 7

Experimente no lugar da linha 60

PLOT INK 1; x,y → e você terá os pontos em azul

Mude depois a linha 60 para

PLOT INK 2; x,y → e você terá os pontos em vermelho.

O seu TK 90X aceita entretanto outras instruções para desenhar e entre elas uma muito importante é a DRAW (palavra chave branca na letra W).

A forma geral deste comando é

DRAW x,y

e que manda o seu TK 90X desenhar (DRAW) uma reta.

A reta vai do último pixel "plotado" até a próxima posição na tela estabelecida por x,y.

Tente o seguinte

```

{ CLS (ENTER)
  DRAW 255,175
}

```

↑ pixels para a direita    ↑ pixels para cima

e veja se sai o que está mostrado na Figura 10.

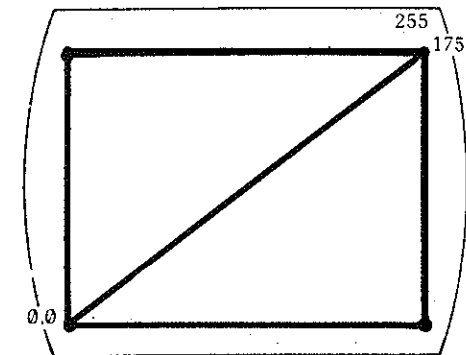
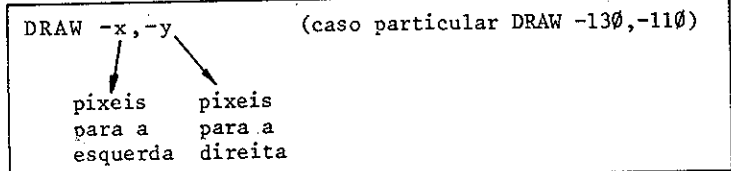


Figura 10

Note que depois da instrução CLS, a reta começará no (0,0) e irá até o outro extremo (na diagonal) ou seja até o ponto (255,175).

Para se desenhar uma linha reta para baixo da tela basta colocar um sinal de menos ("-") na frente do número.

Assim por exemplo, para desenhar uma linha para baixo e para a esquerda basta escrever



Note que se você escrever DRAW -20,20 isto faz com que o cursor gráfico fique 20 unidades para a esquerda e 20 unidades para cima.

Na situação mais comum, se ambos os números após o DRAW são positivos isto quer dizer que o x dá os pixels para a direita e o y os pixels para cima, a partir do último ponto referenciado.

Isto pode parecer um pouco complicado, porém não o é!!!

Assim que você perceber uma vez como se pode desenhar uma figura com a instrução DRAW tudo ficará claro e uma boa maneira para se chegar a este fim é mudar o ponto inicial.

Para tanto, teclé e execute o programa 19.

### PROGRAMA 19

```

10 REM *** desenho de um triângulo ***
20 INPUT "Entre com x = [ ]";x, "Entre com y=[ ]";y
30 PLOT x,y
40 DRAW 30,0
50 DRAW 0,30
60 DRAW -30,-30
70 GOTO 20

```

Você não esqueceu que estes números são obrigatoriamente positivos

$0 \leq x \leq 255$   
 $0 \leq y \leq 175$

Q.I. - Note a vírgula (",") onde você comente esperaria encontrar um ponto e vírgula (";") ou seja na linha 20.

Esta vírgula diz ao TK 90X para imprimir a "próxima coisa" na metade da tela.

Ao executar (RUN) o programa 19 ele lhe perguntará inicialmente pela abscissa (x) e depois pela ordenada (y).

Ele desenhará o pixel e então aplicará a instrução DRAW para desenhar um triângulo a partir deste ponto desenhado pela PLOT.

Caso as coordenadas do seu ponto estejam muito próximas dos bordos da tela para que o triângulo seja desenhado, o TK 90X pode mandar a mensagem:

B : Inteiro excede limite 30 : 1

Se ainda não deu para sentir como funciona o DRAW teclé e execute o seguinte programa:

### PROGRAMA 20

```

10 REM *** reticulado simétrico ***
20 LET p = RND*7 : LET q = RND*7 : LET r = RND*7
30 IF q = r+1 THEN GOTO 20
40 BORDER p : PAPER q : INK r+1
50 FOR j=1 TO 175 STEP 4
60 PLOT 0,j
70 DRAW 255-j,-j
80 PLOT j,0
90 DRAW 255-j,j
100 PLOT 0, 175-j
110 DRAW 255-j,j
120 PLOT j,175
130 DRAW 255-j,-j

```

```

14Ø NEXT j
15Ø FOR z=1 TO 999 : NEXT Z
16Ø CLS
17Ø GOTO 2Ø

```

Execute e veja se sai o que está mostrado.

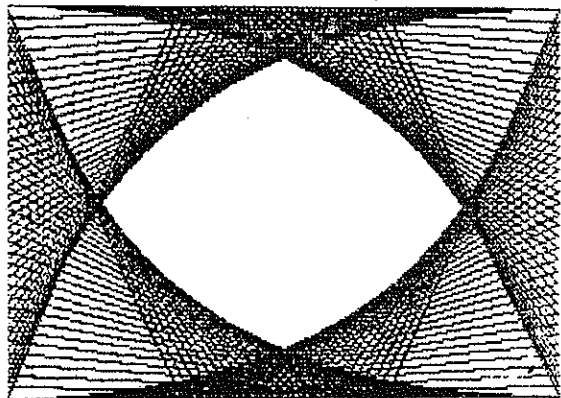


Figura 11

Gostou?

Em caso afirmativo adquira o nosso livro (só faltava ser de um outro...) "Gráficos no TK 9ØX".

Existe uma outra forma, bem mais poderosa de utilizar a instrução DRAW.

Assim como se pode usar DRAW para desenhar linhas retas, pode-se fazer com que o DRAW desenhe partes de uma circunferência ou seja arcos de circunferência bastando para isto acrescentar um terceiro número após a palavra chave DRAW.

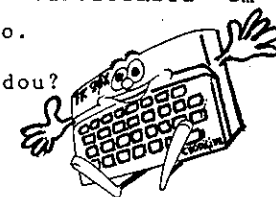
Por enquanto você pode até ficar numa ignorância parcial porém mude por favor a linha 6Ø do programa 19 para:

```
6Ø DRAW -3Ø,-3Ø, PI/2
```

PI é a palavra que está em azul acima da tecla M e como significa  $\pi$  ou seja aproximadamente 3,14 radians o nosso antigo triângulo se transformará em algo parecido com um pedaço de bolo.

Teclou, executou e concordou?

Fantasticooooo!!!



Caso você queira desenhar uma circunferência completa basta utilizar PI.

Mude mais um pouco o programa 19 ou seja acrescente as linhas:

```
4Ø DRAW 8Ø, Ø, PI
```

```
5Ø DRAW -8Ø, Ø, PI
```

e apague a linha 6Ø.

Execute e veja se obtém uma circunferência. Obteve?

Formidável.

Porém, já, já você aprenderá a desenhar circunferências de uma forma bem mais simples.

Como uma última ênfase especial e particular para a instrução DRAW tecle o programa 21.

#### PROGRAMA 21

```

1Ø REM *** o caminho de um microbio desorientado**
2Ø PLOT 127,87
3Ø LET x = 3Ø - RND * 45
4Ø LET y = .3Ø - RND * 45
5Ø LET p = PI * RND
6Ø DRAW x,y,p
7Ø SOUND .25,Ø : SOUND .25,-1 : SOUND .25,-3
   : SOUND .25,-5 : SOUND .25,-7 : SOUND .25,-8
   : SOUND .25,-1Ø : SOUND .25,-12

```

8Ø GOTO 3Ø

Execute e veja se sai, independente da execução diversas vezes da escala musical, algo como mostrado na Figura 12 e até não sair fora da tela.

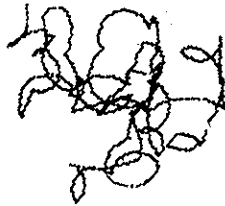
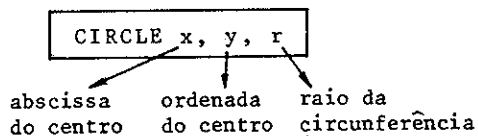


Figura 12

A forma simples que o seu TK 9ØX lhe possibilita desenhar circunferências está ligada a instrução CIRCLE.

A palavra chave CIRCLE é uma palavra em vermelho abaixo da tecla H e a sua forma geral é.



Para começar a entender esta instrução tecle o seguinte programa:

#### PROGRAMA 22

```
1Ø REM *** circunferências ***
2Ø PLOT 75,85
```

```
3Ø DRAW 1ØØ,Ø,PI
4Ø DRAW -1ØØ,Ø,PI
5Ø CIRCLE 15Ø,75,6Ø
```



#### Execute!!!

O que você concluiu?  
Nada!  
Não é verdade!

Você notou que comandado pela instrução CIRCLE a circunferência não só é feita mais depressa como parece mais com uma circunferência...

Execute agora o programa 23 que lhe dará ou melhor encherá a tela de circunferências (25 no total) com centros e raios aleatoriamente escolhidas e ainda com um pequeno "fundo musical".

#### PROGRAMA 23

```
1Ø REM *** circunferências aleatoriamente defini-
das ***
12 PAPER Ø
15 LET n=1
2Ø LET r = RND * 5Ø
3Ø LET x = r + RND * (255 - 2 * r)
4Ø LET y = r + RND * (175 - 2 * r)
5Ø INK (RND * 5Ø) / 7
6Ø CIRCLE x, y, r
7Ø SOUND .25, 9 : SOUND .25, 11 : SOUND .25, 12
: SOUND .25, 5
8Ø LET n = n + 1 : IF n > 25 THEN GOTO 1ØØ
9Ø GOTO 2Ø
1ØØ FOR Z=1 TO 2ØØØ : NEXT Z
11Ø CLS
```



Sai algo como está mostrado na Figura 13.

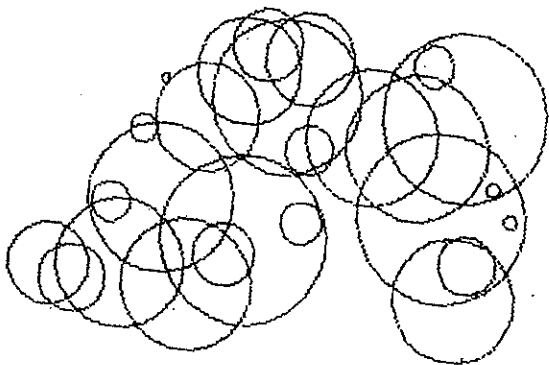


Figura 13

Existe uma outra forma que você caro(a)leitor(leitora) pode usar a instrução DRAW.

Suponha que as coordenadas (posição) do último pixel plotado estão armazenadas em alguma parte da memória do TK 90X.

Desta forma se você deseja usar coordenadas absolutas, assim como quando está verificando uma posição em um mapa geográfico, tudo o que tem que ser feito é perguntar ao TK 90X que coordenadas eram e subtraí-las das novas coordenadas.

Até agora, você sempre pediu ao TK 90X para olhar para os nomes de variáveis nas suas caixas de memória.

Estes nomes eram constituídos de letras ou então de palavras como n\$, s\$, arma, n, etc.

Mas assim como um sítio pode ter um nome (como "Xangrila", "Repouso do guerreiro", "Goiere", etc...) para distinguir um do outro, evitar a confusão e podermos encontrar e entrar no sítio correto

quando somos convidados para algum sensacional churrasco o TK 90X prefere utilizar números no lugar de nomes para identificar locais e principalmente endereços de memória.

O carteiro também gosta de um número grande e bem legível para não atrapalhar o seu trabalho.

O TK 90X possui milhares e milhares de caixas de memória e cada uma delas tem um número particular.

Assim como um carteiro procura desesperadamente o número correto de uma casa ou de um apartamento, também o TK 90X quando perguntado para olhar ou espiar (PEEK) dentro de uma caixa de memória (por exemplo de número 37156) o faz com uma velocidade muito maior do que quando se manda olhar para n\$, "arma", t, m\$ etc.

O único problema é que você deve conhecer o endereço correto.

Como o TK 90X tem muitos endereços (milhares são para uso próprio) e os números correspondentes aos mesmos são bem grandes e nada fáceis para serem memorizados, no Apêndice C do Manual de Operação do TK 90X aparece uma lista completa dos endereços.

Consulte-a!!

O TK 90X armazena a abscissa  $x$  no número de endereço 23677 e a ordenada  $y$  no número de endereço 23678 (verifique isto no Apêndice C do Manual de Operação do TK 90X).

Para fazer com que o TK 90X olhe nessas caixas de memória deve-se utilizar a palavra chave

**PEEK** que é uma instrução em azul acima da letra "O".

A forma mais prática é teclar o programa 24 e assim entender o efeito do PEEK antes de partirmos para uma generalização.

### PROGRAMA 24

```
10 REM *** espiando onde não é simples olhar...***
20 INPUT "Entre com x□"; x; "□Entre com y□"; y
30 DRAW x - PEEK 23677, y - PEEK 23678
40 GOTO 20
```

Quando for executar o programa ao ser "interpelado" pelo TK 90X sobre os valores de x,y entre com os que aparecem ao lado da "casinha" da Figura 11.

Aliás, aos poucos você já está percebendo o que vai se querer programar ou ao menos desenhar.

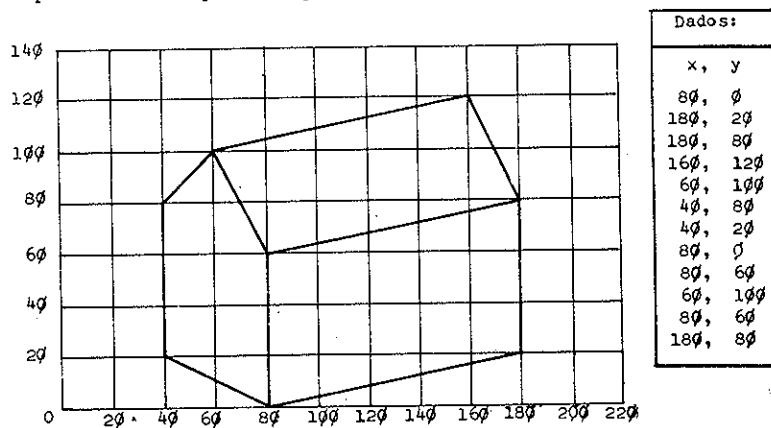


Figura 14

Execute o programa.

Saiu a casinha da Figura 14?

Bom, bom, muito bom.



A sua criatividade está despertada e que tal como exercício intelectual desenhar, aproveitando o programa 24, o avião da Figura 15.

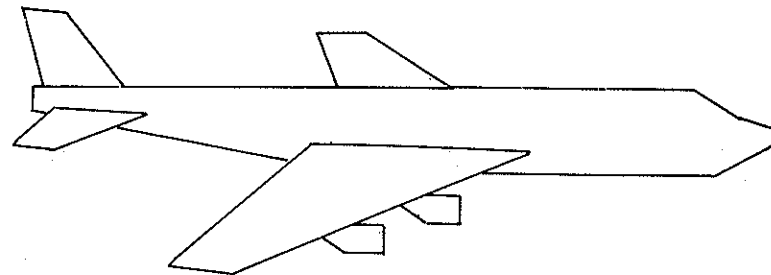


Figura 15

Caso você não consiga, adquira o nosso livro "Gráficos no TK 90X" que lá você encontra a resposta...

Vamos agora indicar-lhe um método melhor para entrar com os números (dados) do que aquele há pouco aplicado ou seja, usando a instrução INPUT.

Você pode armazenar dados (DATA) no seu programa 24 acrescentando as seguintes linhas:

```
50 DATA 80,0,180,20,180,80
60 DATA 160,120,60,100,40,80
70 DATA 40,20,80,0,80,60
80 DATA 60,100,80,60,180,80
```

Você sabe de onde vieram os números acima? Ah, bom!!!

O que se fez anteriormente foi você ler os dados e introduzi-los aos pares através da instrução INPUT dentro do TK 90X.

Caso queira que o TK 90X faça isto para você basta informar-lhe que leia (READ) os dados

(DATA) e para tanto é necessário mudar a linha 20 para

```
20 READ x,y
```

Execute agora o programa 24.

Sem dúvida o TK 90X é bem mais rápido que você!

O TK 90X parou com a mensagem

```
E fim de dados 20:1
```

Agora você já sabe como utilizar todos os comandos gráficos e mais uma vez insisto; obediente leitor(a) tente criar as suas próprias figuras e neste caso é conveniente ter um pouco de papel milimetrado para otimizar o seu trabalho.

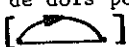
Este capítulo foi farto de novas instruções e um bom resumo ser-lhe-a muito útil.

**CIRCLE x,y,r** → Com esta instrução desenha-se uma circunferência com centro em x,y e um raio de r pixels.


**DRAW x,y** → Diz ao TK 90X para desenhar uma linha reta entre o último ponto "plotado" (PLOT) e o pixel especificado.  
Esta na **tecla W**.

**DRAW x,y,p** → Desenha uma curva do último ponto "plotado" até o pixel especificado por x,y e p é o ângulo especificado em radianos.

Entre dois pontos podemos traçar dois arcos de circunferência diferentes desde que se parta do mesmo ponto.

O arco desenhado no sentido horário é aquele que está acima de dois pontos que unidos estão na horizontal. 



Já o arco desenhado no sentido anti-horário estará abaixo. 

Ângulos com p positivo desenharam arcos no sentido anti-horário e aqueles com p negativo no sentido horário.

**DRAW x-PEEK 23677, y-PEEK 23678** → Desenha uma linha reta do último ponto representado até o novo ponto de coordenadas (x,y).

**INVERSE 1** → Diz ao TK 90X para preencher o pixel na posição especificada com a cor do PAPER ou seja para apagá-lo.  
Está abaixo da **tecla M**

**PLOT x,y** → Diz ao TK 90X para ativar o pixel em uma posição especificada com a cor do INK.  
Está na **tecla Q**

**READ** → Diz ao TK 90X para ler os dados das linhas de instrução DATA que podem estar em qualquer ponto do programa.

**DATA** → Armazena-se aí uma lista de dados.

Caso sejam numéricos basta na linha da instrução DATA ter os dados separados por vírgulas.

Por outro lado se tivermos dados alfanuméricos eles devem aparecer entre aspas.

Os dados nas linhas DATA são lidos e retirados sequencialmente pela instrução READ.

Geralmente por uma questão de organização as instruções DATA aparecem ou no início ou no fim de um programa, porém nada impede que estejam colocadas em qualquer ponto.

Quando a leitura dos dados pela instrução READ



termina, todos os dados deveriam ter sido usados e por isto o número de dados disponíveis deve ser igual ao número de vezes que você indicou, no programa, para que o TK 90X efetue a leitura e a retirada.

Caso a leitura prossiga se produzirá a mensagem de erro

E - Fim dos dados

a menos que você restaure (RESTORE) a lista de dados usando um comando RESTORE.

Cada vez que o RESTORE é executado, os dados das instruções DATA voltam a ser lidos desde o início pela próxima instrução READ. No Capítulo 21 vai aparecer uma super-aplicação do RESTORE (programa 25).

**SAVE "nome" SCREEN\$** → Caso você queira que uma figura fique armazenada em fita basta usar SAVE "nome" SCREEN\$.

**LOAD "nome" SCREEN\$** → Esta é sequência de instruções que você deve enviar ao TK 90X para que ele carregue uma figura para a tela que está guardada na fita.

**PEEK endereço** → Com esta instrução efetua-se a leitura de um determinado endereço da memória, retornando o dado que aí se encontra. Consulte o Apêndice C do seu Manual de Operação do TK 90X.

Portanto com o PEEK podemos manipular a memória do TK 90X.

Que resumo hein!?!?!



É uma ótima idéia ir ao Capítulo 23 e analisar os programas 8, 9, 10, 11 e 12 que serão muito úteis para que você compreenda melhor ainda algumas das novas instruções que surgiram só neste capítulo misturadas obviamente com as que você já conhece melhor.



## Capítulo 21

### Incrementando ao máximo a alta resolução

Agora que você já sabe como se pode desenhar quadros no TK 90X seria uma boa idéia utilizar este recurso em jogos.

Apesar de "muito clássico" vamos começar com o jogo da força o qual nos permite dar-lhe uma série de explicações.

No jogo da força (programa 25) você deve advinhar uma palavra escolhida de uma lista de palavras pelo seu TK 90X.

A medida que você vai errando a figura de um boneco enforcado vai sendo desenhada.

Como este programa é um pouco comprido vamos subdividi-lo em pequenas secções.

Cada trecho pode ser escrito isoladamente.

Uma maneira de fazer isto é usar a instrução GOSUB com a qual se desvia para uma subrotina.

Como você já sabe GOSUB é bem semelhante ao GOTO (aliás você já viu o que faz o GOSUB anteriormente...).

A principal diferença é que com GOSUB o TK 90X lembra onde estava antes no programa.

Ele sai do programa principal, executa o que deve ser feito na subrotina e volta ao programa principal na instrução seguinte àquela da qual ele

saiu toda vez que encontrar a instrução RETURN dentro da subrotina (veja a Figura 16).

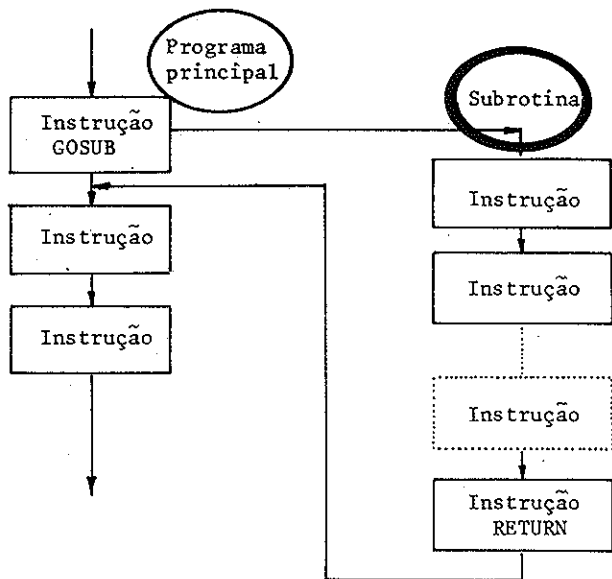


Figura 16

Como é possível repartir um programa em diversas subrotinas?

Um bom início (as vezes) é esboçar o programa todo em um fluxograma como o indicado na Figura 17.

As vezes, por incrível que possa parecer é bom começar pelo fim...

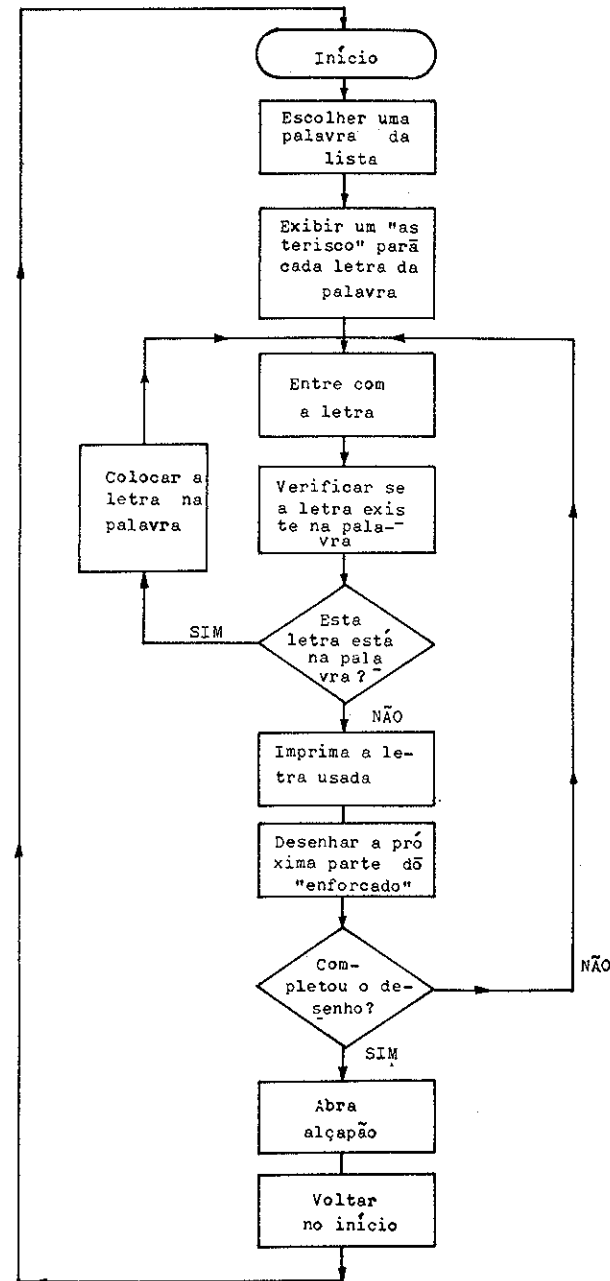


Figura 17

Muitos programas já tem embutido dentro de si o comando SAVE.

Aliás esta é uma outra boa idéia visto que garante a você uma respeitável economia e não leva os seus dedinhos a ter calos inúteis pois não é necessário bater os programas novamente.

As primeiras e neste caso últimas linhas do programa 25 "O JOGO DA FORÇA" são: '

### PROGRAMA 25

```
900 REM *** armazene a rotina
910 SAVE "O jogo da força"
920 CLS : SOUND 1,20
930 PRINT "Reenrole para verificar"
940 VERIFY "O jogo da força"
```

Com estas linhas no programa 25 você pode "preservar" (SAVE) o mesmo usando em algum lugar a instrução GOTO 900.

Quando o programa terminar se ouvirá um som e lhe será pedido cuidadoso(a) leitor(leitora) para verificar (VERIFY) a gravação para estar seguro que realmente se armazenou na fita adequadamente sem nenhuma "mancada" como por exemplo ter ligado o "plug" do gravador no seu TK 90X.

### A próxima rotina é a da lista de palavras

(20 no nosso caso particular)

```
800 REM ** Escolha de uma palavra **
810 RESTORE 865
820 READ número
830 FOR J=1 TO INT (RND*número) + 1
840 READ p$
```

```
850 NEXT J
860 RETURN
865 DATA 20 : REM este é o número de palavras do
      jogo
870 DATA "hidra", "queijo", "crescer", "contumaz",
      "fixar"
875 DATA "ululante", "hebdomadário", "curso",
      "amplo", "xa"
880 DATA "veemência", "discussão", "rural", "giz",
      "ventilador"
885 DATA "lista", "bandeira", "caixa", "luz",
      "persiana"
```

Vejam como funciona este trecho



Linha 810 → Diz ao TK 90X que os dados (DATA) para esta rotina começam na linha 865, ou seja toda vez que se entrar na linha 810 serão lidos todos os dados que forem indicados na linha 830 a partir do primeiro dado da linha 870

Linha 820 → Lê (READ) o número que está armazenado na linha 865.

Este é o número de palavras na lista.

Caso você queira acrescentar mais palavras, pode desde que mude o número que aparece na linha 865.

Linhas 830 a 850 → Com este "loop" escolhe-se uma palavra aleatoriamente da lista e armazena-se a mesma em p\$.

Linha 860 → Faz com que o TK 90X retorne (RETURN) ao programa principal, uma instrução após aquela que fez com que se entrasse nesta subrotina.

O.I. - Se você quiser testar a rotina acima basta escrever GOSUB 800, seguida de um PRINT p\$.

Não esqueça de guardar (SAVE) o programa depois que terminar cada rotina.

Vamos agora mostrar como se pode imprimir asteriscos em número igual ao das letras da palavra escolhida pelo TK 90X.

```

400 REM *** imprimindo asteriscos ***
405 CLS : LET t = 0 : BORDER 3 : PAPER 0 : INK 1
410 PRINT AT 0,10; FLASH 1; "O JOGO DA FORCA"
415 GOSUB 800 : REM Aí ocorre o desvio para a
      subrotina na qual se escolhe uma palavra.
420 LET c$ = ""
425 FOR j=1 TO LEN p$
430 PRINT AT 5,18+j; "*"
435 LET c$ = c$ + "*"
440 NEXT j
445 STOP

```



Vamos explicar um pouco como funciona este trecho

Linhas 405 e 410 → Estas são as primeiras linhas do programa principal.

Com elas se ajustam as diversas cores, o número de tentativas (valor inicial zero) e o título do programa piscante "O JOGO DA FORCA".

Linha 415 → Vai-se a subrotina.

Linha 420 → Você estará mantendo uma cópia da palavra na tela na caixa de memória (variável) c\$.

Inicialmente atribui-se a c\$ uma STRING vazia ou seja o nada para que você tenha certeza

que não existe aí nenhuma palavra antiga e indesejada.

Linhas 425 a 440 → Forma-se um laço com o qual pode-se imprimir os asteriscos, um para cada letra da palavra que você deve adivinhar.

Estes asteriscos serão colocados na caixa de memória (variável) de nome c\$.

Na linha 425 aparece a instrução LEN que diz ao TK 90X para contar quantas letras existem na palavra.

Na linha 430 imprime-se um asterisco cada vez que se dá uma volta - um asterisco para cada letra de p\$.

Neste ponto você pode fazer um teste do seu programa executando-o (RUN) até este ponto.

Como você pode notar surge um certo número de asteriscos no canto da direita da tela.

Agora teclie PRINT p\$ e verifique que existe o mesmo número de asteriscos e de letras.



A seguir vai o trecho do programa que recebe a sua letra ou seja a letra escolhida por você.

```

700 REM ** aceitando a sua letra **
710 PRINT AT 21,0; "Entre com a sua escolha  "
715 PAUSE 0 : SOUND .04,20
720 LET l$ = INKEY$
725 IF l$ = CHR$ 13 THEN GOTO 715
730 PRINT AT 21,25; l$
735 PAUSE 0 : SOUND .04,20
740 LET e$ = INKEY$
745 IF e$ < > CHR$ 13 THEN LET l$ = e$ : GOTO 730

```





```

455 LET  correto = 0
460 FOR i = 1 TO LEN p$
465 IF l$ < > p$ (i) THEN GOTO 490
470 PRINT AT 5,18+i; l$
475 LET  correto = 1
480 SOUND .05,10
485 LET c$(i) = l$
490 NEXT i

```

Mais uma vez, você pode testar o programa até este ponto.

Como anteriormente, o TK 90X escolherá uma palavra e mostrará o número de asteriscos que correspondem as letras da mesma.

Aí, lhe pedirá uma letra, verificará se a mesma está contida na palavra e em caso positivo imprimirá aquela letra nas posições corretas da palavra.

Agora deve-se ensinar ao TK 90X o que fazer se a letra que você escolher estiver errada.

Acrescente ao programa as seguintes linhas:

```

495 IF correto = 0 THEN GOSUB 650
500 IF c$ < > p$ AND t < 10 THEN GOTO 445
510 IF c$ < > p$ THEN GOTO 560

```

Vejamos como funciona o trecho do programa que vai da linha 445 até 510.

Linha 445 → Vai-se a subrotina que pede a você uma letra.

Linha 455 → Aqui a variável "correto" poderá ter um dos dois possíveis valores 0 ou 1.

Se (IF) o seu valor for 1 então "correto" é verdadeira e se (IF) for 0 então (THEN) "correto" é falsa.

Deve-se começar com "correto" valendo inicialmente zero (0).

Linha 460 → Aí começa o "loop" para o número de letras que existe na palavra armazenada em p\$.

Linha 465 → Com esta instrução pode-se saltar o resto do laço se l\$ não é igual a nenhuma letra de p\$. Digamos que p\$ = "giz" então p\$(1) = "g", p\$(2) = "i" e p\$(3) = "z".

Se a sua primeira escolha for por exemplo a letra "a" então l\$ = "a" e a sequência do programa será três vezes as linhas 460, 465 e 490.

Linha 470 → Com esta instrução apresenta-se a letra armazenada em l\$ (e que existe na palavra guardada em p\$) mostrando-se a mesma na tela na posição correta.

Linha 475 → A partir deste ponto "correto" é verdadeira pois a letra é a certa...

Linha 480 → Fazemos um pouco de barulho para comemorar o acerto...

Linha 485 → Mantém na "cópia" c\$ a situação mais recente da tela.

Linha 495 → Faz o programa desviar para uma subrotina que não lhe foi apresentada ainda.

Linha 500 → Envia o TK 90X de volta ao programa se (IF) ambos c\$ não é igual a p\$ e (AND) se o número de tentativas (t) é menor que 10.

Linha 510 → Agora você já deu dez "chutes" e não conseguiu acertar.

Se (IF) c\$ não é igual a p\$ então (THEN) com es



ta instrução o TK 90X é desviado para a rotina que irá enforcar um ser humano... Também esta parte, está logo aí na frente.

### Vamos então completar o programa

Aí vai a subrotina que se chamou na linha 495

```
650 REM imprime-se a adivinhação errada***
655 LET t = t+1
660 PRINT AT 6+t, 21; t; "□"; 2$
665 REM ** Por enquanto vamos deixar uma linha vazia ***
670 RETURN
```

Esta rotina é bem simples de entender.

Com ela conta-se o número de tentativas (linha 655) e utiliza-se esta contagem para fazer na tela uma lista na vertical do número da tentativa e da letra usada na mesma.

Aí então volta-se ao programa principal.

Por enquanto na linha 665 vai uma linha com um REM porém é aí que vai uma das partes mais importantes do programa.

Existem ainda duas pequenas rotinas que você deve teclar e aí o programa todo estará pronto para ser testado.

### Acrescente as linhas

```
520 REM *** Você é um homem ou uma mulher livre***
530 PRINT AT 21,0; "Você está livre!!!"
540 SOUND 2,20 : SOUND 1,20 : SOUND .5,20
550 GOTO 400
560 REM *** A execução correu ***
```

```
570 PRINT AT 21,0; "Lamentavelmente você já era..."
580 SOUND 2,-20 : SOUND 2,-30 : SOUND .5,-40
590 GOTO 400
```

Você pode agora testar o programa e ele deveria (deveria...) funcionar adequadamente!?!?!?

O TK 90X deve escolher uma palavra e exibir o número correto de asteriscos.

Então ele lhe pede uma letra.

Esta é verificada e caso exista na palavra isto lhe é exibido.

Em caso contrário conta-se a sua tentativa errada.

Depois de 10 tentativas erradas, você verificará que foi enforcado(a), porém isto não é tão fácil assim e se usar um pouco a cuca, tiver um pouco de sorte e com todas estas tentativas (10) seguramente escapará!!

Tanto numa situação como na outra o programa recomeçará "automaticamente".

Tudo está funcionando adequadamente?

Uootimo!!!

Escreva então a última parte do programa que consiste de 10 subrotinas, uma para cada adivinhação errada.

De saída, substitua a linha 665 por:

```
665 GOSUB t * 25
```

e aí vai o final:

```
10 GOTO 400
25 REM *** pode desenhar a escora vertical"
30 PRINT AT 6+t,0; "escora vertical"
35 RETURN
```



```

50 REM***pode desenhar a travessa (peça transver-
    sal)***
55 PRINT AT 6+t,0; "travessa"
60 RETURN
75 REM***pode desenhar o suporte***
80 PRINT AT 6+t,0; "suporte"
85 RETURN
100 REM***pode desenhar o nariz***
105 PRINT AT 6+t,0; "nariz"
110 RETURN
125 REM***pode desenhar o rosto***
130 PRINT AT 6+t,0; "rosto"
135 RETURN
150 REM***pode desenhar o corpo***
155 PRINT AT 6+t,0; "corpo"
160 RETURN
175 REM***pode desenhar o braço direito***
180 PRINT AT 6+t,0; "braço direito"
185 RETURN
200 REM***pode desenhar o braço esquerdo***
205 PRINT AT 6+t,0; "braço esquerdo"
210 RETURN
225 REM***pode desenhar a perna direita***
230 PRINT AT 6+t,0; "perna direita"
235 RETURN
250 REM***pode desenhar a perna esquerda***
255 PRINT AT 6+t,0; "perna esquerda"
260 RETURN

```

Você ainda lembra como é chamada cada uma destas subrotinas?

Resp.: Com a linha 665, onde está a instrução GOSUB t\*25.

Em cada tentativa fracassada temos o valor de t e assim por exemplo, no quinto erro o desvio será para 125 e ao se atingir a última "chance" ou seja com t=10 se desviará o programa, devido a esta instrução, para a linha 250.

Agora, quando você executar (RUN) o programa sairá impressa uma lista das "partes" de cima para baixo no lado esquerdo da tela.

O programa está completo e deve funcionar bem...

A única restrição é que ele imprime palavras onde deveriam estar desenhos.

Não vamos deixar você totalmente perdido.

Aí vai uma sugestão - Figura 18 - de um possível desenho.

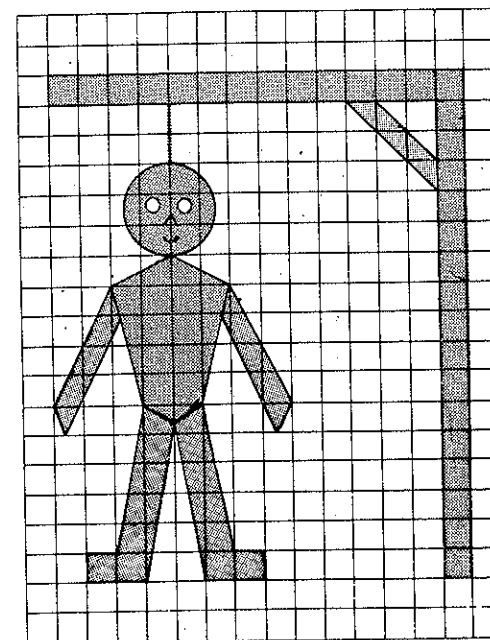


Figura 18

Caso você queira que saia o desenho aí está uma sugestão de como deve ser o seu programa:

```

25 REM *** escora vertical ***
28 RESTORE 44
32 READ x,y : PLOT x,y
34 FOR i = 1 TO 5
36 READ x,y
38 DRAW x - PEEK 23677, y - PEEK 23678
40 NEXT i
42 RETURN
44 DATA 0,15,126,15,103,15,103,144,112,144,112,15

```



Não esqueça de tirar a instrução RETURN da linha 35.

Encaixe, este trecho e veja como ficou o seu programa. Precisa de algum conserto?

Se você quiser que o seu programa comece a funcionar automaticamente basta acrescentar

LINE 400

à linha 910 ou seja escrever

```
910 SAVE "O jogo da forca" LINE 400
```

Você encontrará a instrução LINE impressa na tecla vermelha abaixo da tecla 3.

Agora guarde (SAVE) o seu programa com um GOTO 900.

Na próxima vez que você carregá-lo (LOAD) ele saltará direto para a linha 400 sem esperar pelo seu RUN.

É realmente um dever seu completar os outros desenhos e aí sim se promova, pois você é de fato um(a) fora de série...

Mostre para os pais, para os(as) amigos(as) e principalmente aos(as) inimigos(as).

As novas instruções ou as já conhecidas que devemos esmiuçar um pouco mais são:

AND (linha 500) Uma grande simplificação ocorre no programa quando utilizamos os operadores lógicos (Tabela 5).

Operadores lógicos	
AND (e)	→ uma relação e (AND) outra relação
OR (ou)	→ uma relação ou (OR) outra relação
NOT (não)	→ relação não (NOT)

Tabela 5

Assim como "<=", ">=" e "<>", AND, OR e NOT têm teclas próprias *não sendo* necessário digitá-las caractere por caractere.

Havendo a possibilidade de utilizar os operadores lógicos, em uma única instrução podem ser feitas várias comparações ao mesmo tempo.

Pode-se na mesma instrução ter mais de um operador lógico e inclusive uma mistura dos mesmos.

CHR\$ (linhas 725 e 745) O conjunto de caracteres utilizado pelo TK 90X é em sua grande maioria o mesmo utilizado em outros micros e que se chama ASCII (American Standard Code Information Interchange).

Além dos caracteres chamados "padrões" tais como os dígitos, o alfabeto (com maiúsculas e minúsculas), símbolos comuns como (\$, <>, <=, etc) o conjunto de caracteres do TK 90X, inclui as palavras chave (IF, GOTO, READ, etc.)

Isto não é tudo, pois dentro dos 256 caracteres disponíveis o seu TK 90X possui dois conjuntos de caracteres especiais, utilizados amplamente nos idiomas português e espanhol (consulte o Capítulo 18 do seu Manual de Operação do TK 90X).

Para maior comodidade do(a) usuário(a) o TK 90X dispõe de um Editor de UDG que permite a você definir caracteres inéditos (veja os programas 28, 29 e 30 que destacarão a sua genialidade).

Bem, é a função CHR\$, que vem normalmente precedida do comando PRINT, que é empregada para obter o caractere correspondente ao número que lhe serve de argumento.

Tecla `PRINT CHR$ 56` e veja que o TK 90X lhe devolve o número 8.

Vá ao Apêndice D do Manual de Operação do TK 90X para confirmar isto.

Por outro lado se você teclar no modo imediato

`PRINT CHR$ 86`

o TK 90X lhe mostrará a letra V.

Execute o seguinte *mini-programa* para "sentir" todo o Apêndice D.

#### PROGRAMA 26

```
5 REM *** A função CHR$ ***
10 FOR j = 32 TO 255
20 PRINT "CHR$"; j; " = "; CHR$ j; " ";
30 NEXT j
```

Execute e compare a saída com o que tem no Apêndice D do Manual de Operação do TK 90X.

Bateu, não é?



#### Conclusão:



A função CHR\$ retorna o caractere correspondente ao código.

Deve estar entre 0 e 255.

Aproveitando a oportunidade vamos falar um pouco da função CODE que vai aparecer no programa 27.

A função CODE se aplica a uma STRING e retorna o código ASCII (veja o Apêndice D do Manual de Operação do TK 90X) do primeiro caractere da sequência alfanumérica.

Assim por exemplo se você teclar

`PRINT CODE "Victor"`

o TK 90X lhe exibirá o número 86 que corresponde a letra V no código ASCII.

A função CODE está acima da `tecla I`.

#### O.I. zão

Até agora, usamos e abusamos da linguagem BASIC porém não é só isto que o seu TK 90X entende.

Pode-se escrever programas mais excitantes ainda e principalmente executados mais rapidamente pelo TK 90X se os mesmos forem escritos em linguagem de máquina (do TK 90X).

Aliás esta é a linguagem do microprocessador do TK 90X.

Como um pequeno exemplo aí vai o programa 27 no qual você poderá comparar a velocidade de ambos.

Logo, logo você verá alguns textos de nossa autoria em linguagem de máquina do TK 90X.

Bem aí vai o programa:

## PROGRAMA 27

```

10 REM *** carregando em hexadecimal ***
20 CLEAR 31999: LET i = 32000
30 READ c$: IF c$ = "STOP" THEN STOP
    Este é o comando STOP
50 LET n = CODE c$ : LET x = 16*((n-48
    AND n > 47 AND n < 58) + (n-55
    AND n > 64 AND n < 71))
60 LET n = CODE c$(2) : LET x = x +
    (n - 48 AND n > 47 AND n < 58) +
    (n - 55 AND n > 64 AND n < 71)
70 POKE i, x: LET i = i + 1
80 LET c$ = c$(3 TO): IF c$ = "" THEN GOTO 30
90 GOTO 50
100 REM ** dados em linguagem de máquina **
110 DATA "01FF00": REM LD BC,255
120 DATA "3E40": REM LD A,40
130 DATA "D7": REM RST 10
140 DATA "10 FB": REM DJNZ-5
150 DATA "C9": REM RET
160 DATA "STOP"
500 REM ** programa equivalente em BASIC **
510 FOR I = 255 TO 0 STEP -1
520 PRINT CHR$(64);
530 NEXT I
1000 REM ** Chamar a linguagem de máquina **
1010 PRINT USR 32000

```



Para sentir a diferença entre um programa em linguagem BASIC e o mesmo em linguagem de máquina escreva RUN 500 e tecla **ENTER**.

Dá para notar a diferença de velocidade?  
Sem dúvida!!! Haja @ não!?!?!?

Não vamos entrar em grandes detalhes no momento, sobre o programa 27 porém convém "clarear" um pouco duas instruções.

**CLEAR** (linha 20) → Diz ao TK 90X para reservar algum espaço para a linguagem de máquina. Esta instrução está sobre a **tecla X**. Com CLEAR apenas, são zeradas as variáveis BASIC existentes na memória. Caso, se tenha o formato **CLEAR ramtop**, ramtop define o maior endereço do BASIC.

**USR** (linha 1010) → Diz ao TK 90X para executar a rotina em linguagem de máquina começando na posição de memória especificada. Está acima da **tecla L**.

Já, já você vai se maravilhar mais ainda com o seu TK 90X quando utilizar **POKE**, **USR** e **BIN**.

Não se afobe leia tudo com atenção e se for possível duas vezes...



O TK 90X fica até mais alegre e "esperto" quando você consegue conversar com ele na linguagem de máquina. Aqui terminou o "oizão" e restam os seguintes comentários finais sobre o programa 25.

**LEN** (linhas 425 e 460) → Retorna o número de caracteres existentes na STRING especificada entre aspas ou armazenada na variável STRING.

**LINE** (linha 910) → Diz ao TK 90X para começar o programa assim que o mesmo terminar de ser carregado.

**RESTORE** (linha 810) → Restaura os valores contidos nas linhas DATA permitindo nova leitura pelo comando READ. Quando seguido de um número de linha (no nosso caso tem-se RESTORE 865) inicia a busca dos dados a partir daquela linha.

Quando o TK 90X é ligado ele conhece a forma de 111 caracteres.

Este é o número total de letras, números, sinais de pontuação e símbolos gráficos impressos nas teclas.

O TK 90X tem também 21 caracteres que facilmente podem ser "redesenhados".

Estas são cópias das 21 primeiras letras de forma (de A até U).

Digite as letras de A até U em modo gráfico, isto é comece com PRINT "(em modo K e feche as aspas em modo L)".

**PRINT "ABCDEFGHIJKLMNOPQRSTU"**

O resultado será a impressão destas letras na forma maiúscula.

Pouco interessaria isto se o conjunto obtido em **G** fosse idêntico ao conseguido por **CAPS SHIFT**.

Na realidade estas letras estão só esperando para serem substituídas pelos símbolos gráficos que você criativo(a) leitor(a) vai inventar.

Mas antes de criar para você alguns destes caracteres, vou ensinar-lhe a acessar os caracteres especiais já existentes.

Digite:

UDG Ø **ENTER** [ se obtém UDG com **SYMBOL SHIFT**



e **tecla X** e UDG vem de user definable graphics, isto é, caracteres gráficos definidos pelo usuário(a)].

Repita agora o PRINT", entre no modo **G** e digite cada letra de A-U novamente deixando um espaço em branco entre cada uma.

Esta vez o resultado deve ser

**Á É Í Ó Ū ā ē ĩ ó ú Æ Ê â ê ã ç ç**

Viva, viva, viva o TK 90X!!!

Finalmente vamos poder escrever em Português.

**O.I.** - Se você quiser acessar o conjunto de particulares caracteres do espanhol repita o procedimento anterior porém escreva UDG 1.

É muito interessante o resultado final. Experimente!?!?!

Você agora pode "escrever corretamente "português" e para tanto faça o seguinte:

UDG Ø : PRINT "português"



e para obter o "ê" use a tecla P no modo gráfico ou seja tecle (< CAPS SHIFT > < 9 >).

Para conhecer outras novidades tecle inicialmente o programa 28.

### PROGRAMA 28

```

10 REM *** Conhecendo U.D.G.***
20 PRINT CHR$ 144
30 FOR j=0 TO 7
40 READ byte
50 POKE j +USR CHR$ 144, byte
60 NEXT j
70 PRINT CHR$ 144
80 DATA BIN 11111111
90 DATA BIN 10011001
100 DATA BIN 10011001
110 DATA BIN 10011001
120 DATA BIN 11111111
130 DATA BIN 10011001
140 DATA BIN 10011001
150 DATA BIN 11111111

```


É aqui que se chama a instrução UDG. Com ela se inserem caracteres especiais. Como comentamos há pouco o código que vem depois da instrução UDG pode ser 0 para caracteres em Português, 1 para caracteres em Espanhol e 2 para o editor UDG

Cada caractere é formado de 64 pixels.

Eles compõem uma matriz (veja o Capítulo 22) ou seja um tabuleiro de xadrez de 8 linhas por 8 colunas.

Cada linha de instrução DATA é constituída agora apenas de "uns" e "zeros".

Dá para você notar o tipo de padrão que os "uns" fizeram?

É algo deste tipo  , não?

### Explicação sobre o programa 28

Linhas 20 e 70 — Ambas dizem ao TK 90X para desenhar o mesmo caractere, CHR\$ 144.

Agora execute o programa.

Você viu o que aconteceu?

Inicialmente a linha 20 imprimiu o caractere 144.

Depois de ter sido feito o "loop" imprimiu-se de novo o caractere 144.

Porém desta vez não foi mais a letra maiúscula "A" e sim um quadrado com uma cruz dentro.

Agora ficou evidente, creio eu, reconhecer este desenho (padrão) das linhas de DATA.



Entrar com CHR\$ 144 é o mesmo que dizer ao TK 90X para imprimir "A" (PRINT "A") onde "A" significa entre em modo gráfico apertando a sequência CAPS SHIFT , a tecla 9 , a tecla A e finalmente a tecla 9 .

Para cada "1" na linha de DATA, o pixel correspondente da matriz 8x8 é ativado e com esta simples explicação você pode desenhar, a partir de agora, os mais sugestivos caracteres como por exemplo os que irão aparecer no programa 29 que podemos chamar de "Invasores".

### PROGRAMA 29

Tecla primeiro todo o programa e guarde na fita com o nome "Invasores".

```

5 REM *** Invasores ***

```

```

10 GOSUB 800 : REM**É nesta subrotina que se definiem os caracteres**

```

```

20 LET a$ = "ABABABAB"
30 LET b$ = "BABABABA"
40 LET c$ = "   " 32 espaços em branco
50 LET r$ = "C"
60 PAPER 0 : INK 6 : BORDER 0 : CLS
70 LET arma = 1 : PRINT AT 21, arma; r$
80 LET invasor = 0
90 GOTO 500
100 REM *** teclas para mexer a sua arma ***
110 LET m$ = INKEY$
120 IF m$ = "" THEN SOUND .125, -40+j : GOTO 180
130 IF m$ = "0" THEN GOSUB 200
140 PRINT AT 21, arma; " "
150 IF m$ = "8" AND arma < 29 THEN LET arma=arma+1
160 IF m$ = "5" AND arma > 0 THEN LET arma=arma-1
170 PRINT AT 21, arma; r$
180 SOUND .004, -30+j
190 RETURN
200 REM *** disparar a arma***
210 FOR l = 20 TO j STEP -1
220 IF SCREEN$ (l, arma) = "" THEN GOSUB 300
230 PRINT AT l, arma; "!"
250 SOUND .00125, l
260 PRINT AT l, arma; " "
270 NEXT l
280 RETURN
300 REM***mostrando o invasor atingido***
310 BORDER 7
320 LET a$ (arma+1) = " "
330 LET b$ (arma+1) = " "
340 SOUND .25, 40 : SOUND .125, 30
350 LET invasor = invasor+1
360 BORDER 0
370 PRINT#1, AT 0, 7; " "invasores atingidos"; inva
sor

```

Deixe entre cada caracter gráfico três espaços em branco

```

330 RETURN
500 REM***Aí vai o laço principal***
510 FOR j=0 TO 20
520 PRINT AT j, 0; a$
530 GOSUB 100
540 PRINT AT j, 0; b$
550 GOSUB 100
560 PRINT AT j, 0; c$
570 NEXT j
580 REM**contagem e repetição**
590 PRINT AT 2, 5; "Você acertou "; invasor;
" dos 8 existentes"
600 PRINT AT 7, 3; FLASH 1; "Aperte qualquer tecla
para brincar de novo"
610 PAUSE 0
620 RUN
800 REM***Aí vão os desenhos dos caracteres gráfi
cos***
810 READ c$: IF c$ = "f" THEN RETURN
820 REM ** Para cada caractere 8 bytes**
830 FOR j=0 TO 7
840 READ byte
850 POKE j+USR c$, byte
860 NEXT j
870 GOTO 810
880 DATA "a"
890 DATA BIN 00000000
900 DATA BIN 00111100
910 DATA BIN 01011010
920 DATA BIN 01111110
930 DATA BIN 00111100
940 DATA BIN 00100100
950 DATA BIN 01000100
960 DATA BIN 10000100

```

```

970 DATA "b"
980 DATA BIN 00000000
990 DATA BIN 00111100
1000 DATA BIN 01011010
1010 DATA BIN 01111110
1020 DATA BIN 00111100
1030 DATA BIN 00100100
1040 DATA BIN 00100010
1050 DATA BIN 00100001
1060 DATA "c"
1070 DATA BIN 00000000
1080 DATA BIN 00011000
1090 DATA BIN 00011000
1100 DATA BIN 00111100
1110 DATA BIN 00011000
1120 DATA BIN 00111100
1130 DATA BIN 11111111
1140 DATA BIN 11111111
1150 DATA "f"

```

Bem aí está todo o programa 29 "Invasores"

Antes de executá-lo leia algumas explicações.

Neste jogo você tem oito invasores e uma arma espacial e deve exterminá-los!!!

O símbolo em negrito é para destacar o modo gráfico e a letra da tecla é por exemplo **A**.

Mais uma vez destaque que nas linhas 20 e 30 devem existir três espaços em branco entre cada caractere gráfico.

No trecho que vai da linha 20 até a linha 90 ajustaram-se os valores para as cores, para as variáveis e principalmente atribuíram-se valores as variáveis STRING a\$, b\$, c\$ e r\$.

No trecho que vai da linha 100 até a linha 190 tem-se a subrotina que lê o teclado e move a arma.

No trecho que vai da linha 200 até a linha 280 tem-se a subrotina que verifica se o míssil está na eminência de bater em algo.

SCREEN\$ não pode reconhecer um UDG ou seja um caractere criado por você estimado(a) usuário(a).

Quando encontra um UDG ela (a instrução SCREEN\$) retorna uma STRING vazia e é nisto que se baseia o teste; ver quando é que se encontra o "nada"...

Esta subrotina desloca também o projétil (veja a linha 230).

No trecho que vai da linha 300 até a linha 380 tem-se a subrotina que é chamada quando um invasor é atingido.

As linhas 320 e 330 são para apagar o invasor substituindo o seu caractere em a\$ e b\$ com um espaço em branco.

Esta rotina acompanha também a evolução da eliminação dos invasores e imprime-se a contagem na tela no espaço reservado a comunicação com o TK 90X ("espaço de trabalho").

Tenha muito cuidado quando utilizar este espaço da tela, esteja seguro que isto não provocará o roçamento (scroll) da tela ou alguma coisa ímpar ocorra.

No trecho que é composto pelas linhas de 500 a 570 tem-se o "loop" que faz os invasores se movimentarem e onde se chama a subrotina que permite a sua arma se deslocar na tela. Não esqueça que com a **tecla 5** você vai para a esquerda e com a **tecla 8** vai para a direita.

O trecho que vai da linha 580 à 620 lhemos tra a contagem final e permite que se comece tudo de novo caso esta for a sua vontade senhor(a) dono(a) do TK 90X.

É na subrotina que começa em 800, que estão os caracteres especiais criados com o auxílio da notação binária (BIN), colocados na memória do TK 90X com o auxílio da instrução POKE e achados pelo TK 90X, quando solicitado, com a instrução USR.

Ressaltando, não esqueça pois que o conteúdo de cada tecla (de A até U) pode ser trocado por você pois aí tem-se os caracteres UDG ("user definable graphics").

Toda vez entretanto, que o TK 90X for ligado o conteúdo padrão aparecerá.

Se você quiser uma retrospectiva, verá que no "vermelho" UDG obtém-se o conteúdo das teclas A até U em modo G, quando se digita:

```
< PRINT > " < " > G < A > < B > ... < U > "
```

Não confunda com os caracteres que obteve com UDG 0 (ou com UDG 1).

Mais uma vez para entrar no modo gráfico a sequência é:

CAPS SHIFT

e

tecla 9



O. Izão

No livro dos meus filhos Sergio e Victor você poderá sentir toda a importância da existência dos caracteres UDG.

O nome do livro eu quase ia esquecendo...

É "Jogos e Desenhos no TK 90X" vol. 1.

Porém, depois deste comercial não posso ficar em um único exemplo e é por isto que existe o programa 30.

### PROGRAMA 30 "A caça ao submarino"

Você vai participar de um jogo eletrônico que usa UDG.

Neste programa você aprenderá como se pode criar figuras ou formas grandes compostas de vários pequenos UDG ou seja caracteres criadas por VOCÊ PRÓPRIO(A).

No programa 30 os caracteres A, B e C são modificados para no conjunto representarem um destróier e é nas teclas D, E e F que se cria um submarino.

Como no programa 29 você tem que colocar o TK 90X no modo gráfico e aí apertar a tecla requerida.

Bem aí, vai todo o programa "A caça ao submarino"

```
5 REM *** A caça ao submarino ***
10 LET l$ = " "
20 FOR j = 0 TO 2
30 PRINT INK 1; PAPER 5; l$
40 NEXT j
50 FOR j=3 TO 20
60 PRINT INK 1; PAPER 4; l$
70 NEXT j
80 PRINT PAPER 6; l$
90 BORDER 4 : PAPER 8
100 GOSUB 1000
110 LET d$ = "ABC"
```

tecle 32 espaços em branco

```

120 LET s$ = "DEFO"
130 LET dest = 13
140 LET suby = INT(RND*16)+4
150 PRINT AT 2, dest; d$
160 GOTO 500
170 REM***Lançamento de carga de profundidade***
180 FOR h = 3 TO 20
190 IF subx > 0 THEN PRINT AT suby, subx; s$
200 IF subx = 0 THEN PRINT AT suby, 0; "    "
210 IF SCREEN$(h,dest+5) = "" THEN GOTO 310
220 PRINT AT h, dest+5; "O"
230 LET subx = subx-1
240 SOUND .05,-30
250 PRINT AT h,dest+5; " "
260 NEXT h
300 RETURN
310 REM***Acertando o submarino***
320 FOR z=1 TO 13
330 PRINT AT h,dest+5; "█"
340 BORDER 0
350 SOUND .005,RND*(-25)
360 PRINT AT h,dest+5; "█"
370 BORDER 7
380 SOUND .004, RND*(-25)
390 NEXT z
400 BORDER 4
410 PRINT AT suby, subx; "    "
420 GOTO 140
500 REM ***Este é o loop principal***
510 LET subx = 28
520 IF subx>0 THEN PRINT AT suby,subx; s$
530 LET subx = subx-1
540 LET m$ = INKEY$

```

```

550 SOUND .1,-20
560 IF m$ = "0" THEN GOSUB 170
570 IF m$ = "5" AND dest > 1 THEN LET dest = dest-1
580 IF m$ = "8" AND dest < 24 THEN LET dest=dest+1
590 PRINT AT 2, dest; d$
600 IF subx >= 0 THEN GOTO 520
610 PRINT AT suby,0; "    "
620 GOTO 140
1000 REM***Definindo os caracteres***
1010 READ c$ : IF c$ = "u" THEN RETURN
1020 FOR b=0 TO 7
1030 READ byte
1040 POKE b+USR c$, byte
1050 NEXT b
1060 GOTO 1010
1070 DATA "a"
1080 DATA BIN 00000000
1090 DATA BIN 00000001
1100 DATA BIN 00000001
1110 DATA BIN 01111001
1120 DATA BIN 00011001
1130 DATA BIN 11111111
1140 DATA BIN 01111011
1150 DATA BIN 00111111
1160 DATA "b"
1170 DATA BIN 10000000
1180 DATA BIN 10011011
1190 DATA BIN 10011011
1200 DATA BIN 10011011
1210 DATA BIN 11111111
1220 DATA BIN 11111111
1230 DATA BIN 01110111
1240 DATA BIN 11111111

```

Com a tecla 8  
movimenta-se  
o destroyer  
para a direita

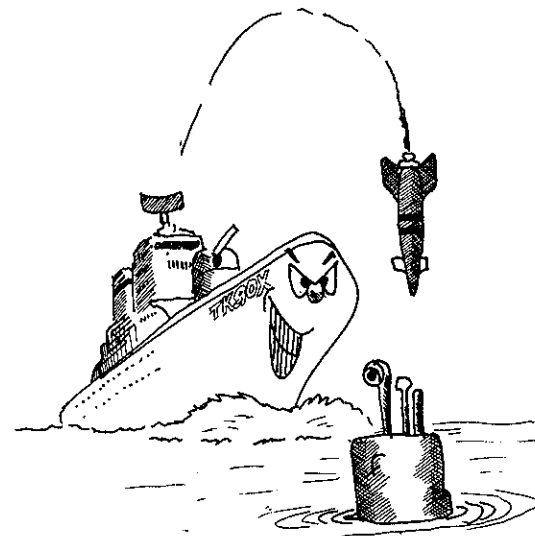
Com a tecla 5 mo-  
vimenta-se o des-  
troier p a a es-  
querda

Lança-se a carga  
ao pressionar a  
tecla 0

125Ø DATA "c"  
 126Ø DATA BIN ØØØØØØØØ  
 127Ø DATA BIN ØØØØØØØØ  
 128Ø DATA BIN ØØØØØØØØ  
 129Ø DATA BIN ØØ1111ØØ  
 130Ø DATA BIN ØØ11ØØØØ  
 131Ø DATA BIN 11111111  
 132Ø DATA BIN Ø111Ø111  
 133Ø DATA BIN 1111111Ø  
 134Ø DATA "d"  
 135Ø DATA BIN ØØØØØØØØ  
 136Ø DATA BIN ØØØØØØØØ  
 137Ø DATA BIN ØØØØØØØØ  
 138Ø DATA BIN ØØØØØØØØ  
 139Ø DATA BIN ØØØØØØØØ  
 140Ø DATA BIN Ø1111111  
 141Ø DATA BIN 11111111  
 142Ø DATA BIN Ø1111111  
 143Ø DATA "e"  
 144Ø DATA BIN Ø11ØØØØØ  
 145Ø DATA BIN ØØ1ØØØØØ  
 146Ø DATA BIN Ø1111ØØØ  
 147Ø DATA BIN Ø1111ØØØ  
 148Ø DATA BIN Ø1111ØØØ  
 149Ø DATA BIN 11111111  
 150Ø DATA BIN 11111111  
 151Ø DATA BIN 11111111  
 152Ø DATA "f"  
 153Ø DATA BIN ØØØØØØØØ  
 154Ø DATA BIN ØØØØØØØØ  
 155Ø DATA BIN ØØØØØØØØ  
 156Ø DATA BIN ØØØØØØØØ  
 157Ø DATA BIN ØØØØØØØØ



158Ø DATA BIN 111111Ø1  
 159Ø DATA BIN 11111111  
 160Ø DATA BIN 111111Ø1  
 161Ø DATA "u"



Antes de brincar com o programa leia as seguintes explicações:

Em primeiro lugar, mais uma vez e agora a última, para conseguir os caracteres A,B,C,D,E e F (linhas 11Ø e 12Ø) não esqueça de colocar o TK 9ØX no modo gráfico (**CAPS SHIFT** e tecla 9 apertadas simultaneamente).

No trecho que vai da linha 5 até a linha 16Ø está a rotina para o ajuste inicial da tela.

O topo da tela simula o céu, depois vem o mar e em seguida ou melhor em baixo um pouco de areia...

Note que na linha 9Ø ao PAPER se deu o valor 8.

Isto não é uma cor!!!

PAPER 8 é transparente como o vidro!!!

Isto permite que se veja o fundo...

Em qualquer impressão posterior se usará a cor para o PAPER que já foi ajustada para aquela parte da tela.

É na linha 100 que se chama a rotina na qual se definiram os caracteres que serão atribuídos a d\$ (linha 110) e s\$ (linha 120) ou seja crian do respectivamente o destróier e o submarino.

A variável "dest" guarda a posição do destróier, enquanto que a variável "suby" indica a profundidade do submarino.

No trecho que vai da linha 170 a 300 está a subrotina que é chamada por você caro(a) possuidor(a) deste livro e do TK 90X quando pressionar a tecla 0.

Ela manipula o movimento do submarino na tela e o movimento para baixo da carga de profundidade.

Na linha 210 verifica-se se alguma coisa foi atingida.

Se isto ocorrer desvia-se para a linha 310.

No trecho que vai da linha 310 até a linha 420 o bordo pisca e simula-se uma explosão.

Finalmente faz-se o bordo voltar a cor original, apaga-se o naufrágio (destruição) e vai-se para uma próxima diversão de afundamento".

No trecho do programa da linha 500 até a 620 está o "loop" principal que entre outras coisas permite movimentar o destróier, disparar a carga e movimentar o submarino.

Finalmente no trecho da linha 1000 até a 1610 está a definição dos caracteres que constituem o destróier (guardados nas letras A, B e C) e o submarino (definidos nas letras D, E e F).

Sai-se automaticamente desta subrotina quando for lido para c\$ o valor "u" de último.

Para que o TK 90X pudesse entender estes dois últimos programas (29 e 30) foram utilizadas as seguintes novas e muito importantes instruções:

**BIN** → Diz ao TK 90X que o número que a segue é um número binário (BINary)

Ela está sobre a **tecla B**.

Com esta instrução um número em binário é convertido para decimal.

Caso o número seja maior que 65535 ocorrerá um erro de sintaxe.

Se por exemplo você escrever

PRINT BIN 10111

o TK 90X lhe retorna o número 23.

Como cada dado é composto de 8 bits ou seja por um byte, para se representar um caracter, por exemplo o da Figura 19, teríamos que "traduzir" a sua forma em bits.

O TK 90X usa a linguagem binária e só entende os números 0 e 1.

Um caractere pode ser definido em uma matriz 8x8 (falaremos bastante de matrizes no Capítulo 22) e portanto para que o TK 90X entenda o caractere da Figura 19 deve-se apresentar a ele os seguintes sequenciais de zeros (0) e uns (1).

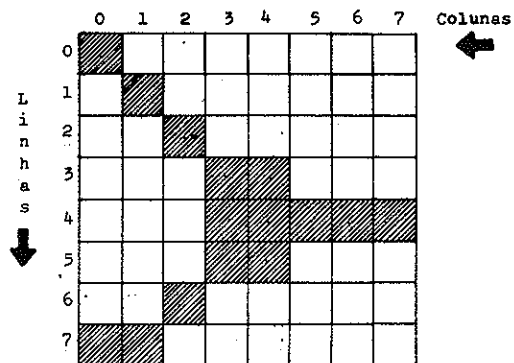


Figura 19

Linha

0	→	1	0	0	0	0	0	0	0
1	→	0	1	0	0	0	0	0	0
2	→	0	0	1	0	0	0	0	0
3	→	0	0	0	1	1	0	0	0
4	→	0	0	0	1	1	1	1	1
5	→	0	0	0	1	1	0	0	0
6	→	0	0	1	0	0	0	0	0
7	→	1	1	0	0	0	0	0	0

Como você vê, é bem fácil criar um caractere especial basta ter um pedaço de papel quadriculado e utilizar quadriculados 8x8.

Aliás, como auxílio para que não perca tempo na localização dos seus desenhos na Figura 20 você tem a tela do seu TK 90X quando estiver trabalhando com a instrução PRINT AT.

Tire quantas cópias quiser da mesma e gane tempo na localização dos seus "engenhos" no momento que se investir com o título e adotar o trabalho de programador(programadora).

Nesta altura dos acontecimentos já dá para programar alguma coisa, não é?

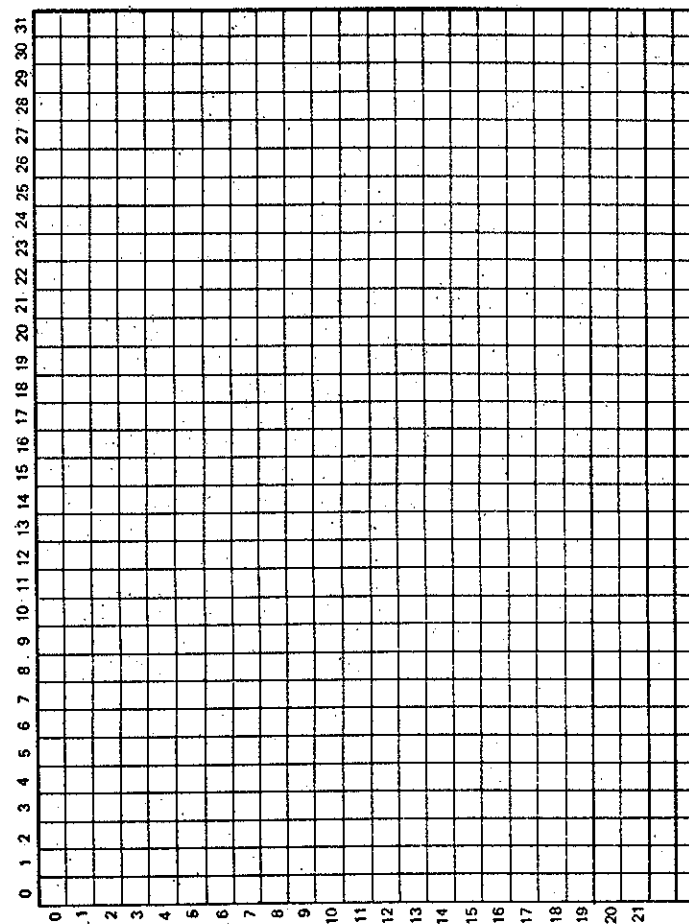


Figura 20 - Representação gráfica da tela de vídeo usado pelo TK 90X



**PRINT # 1** → Diz ao TK 90X para imprimir no "espaço de trabalho".

**POKE** → Diz ao TK 90X para armazenar um inteiro entre 0 e 255 em uma específica posição da memória.

A instrução POKE está na **tecla O**

A forma geral da instrução POKE é

POKE endereço da memória, valor

onde "endereço da memória" pode ser qualquer valor inteiro de 0 a 65535 e "valor" pode ser qualquer número inteiro de 0 a 255.

Caso se apresente algum valor fora desta faixa irá aparecer erro do tipo B.

Assim por exemplo POKE 42384,37 indica a introdução do valor 20 (em decimal) no endereço de memória 22384 (em decimal).

**USR** → Diz ao TK 90X para achar a posição de memória ou endereço para um específico caractere gráfico definido pelo(a) usuário(a).

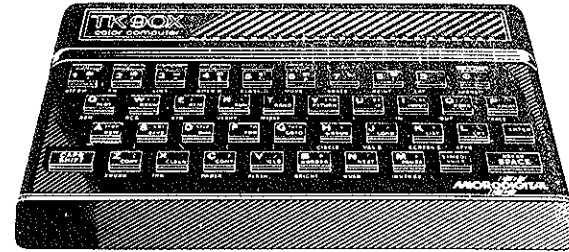
Está na **tecla L**.

De uma forma geral com a instrução **USR endereço de memória** executa-se uma subrotina em linguagem de máquina que se inicia no endereço especificado.

Será que agora você entende um pouco melhor o que se obtém com a instrução da linha 1040 do programa 30?

Acho que sim, porém como auxílio note que b+USR c\$ a cada volta no "loop" permite que se tenha um novo endereço de memória e a cada volta guarda-se aí um valor da variável byte.

Como são 8 voltas tudo é guardado em um caractere ou seja aquele atribuído a c\$.



**TK 90X, o micro cheio de programas.**

# Capítulo 22

## Matrizes

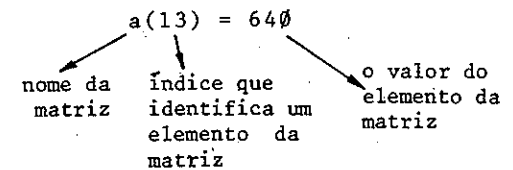
### Definição

Matriz é um conjunto de variáveis com o mesmo nome.

Cada elemento de uma matriz é identificado por um ou mais índices numéricos (de uma forma única) e que aparecem entre parênteses seguindo o nome da matriz.

Em outras palavras quando possuímos uma série de dados inter-relacionados de alguma forma entre si podemos guardá-los em uma única variável que costuma-se chamar de MATRIZ.

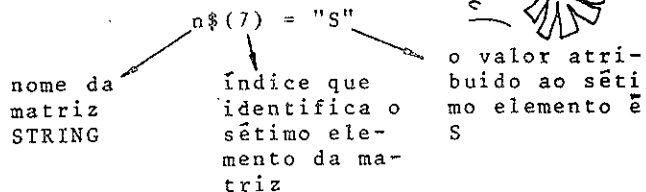
Um exemplo de um elemento de uma matriz unidimensional é



Uma matriz pode conter dados numéricos ou alfanuméricos (STRINGS).

Quando os elementos de uma matriz são dados STRINGS, o nome da mesma deve ser seguido pelo símbolo "\$".

Assim por exemplo



Costuma-se chamar os elementos de uma matriz ("array" em inglês) de *variáveis subscriptas ou indexadas*.

Antes de podermos usar qualquer matriz devemos reservar espaço para ela na memória do seu TK 90X.

É para esta finalidade que existe a instrução DIM (de dimensionamento)

Suponha que você queira montar uma matriz contendo a "coleção" de variáveis  $a_1, a_2, a_3, a_4, a_5, a_6$  e  $a_7$ .

Neste caso em primeiro lugar você deve informar ao TK 90X quantos elementos comporão a matriz  $a()$ .

Isto é feito através da instrução DIM ou seja escrevendo logo no início do programa

```
DIM a(7)
```

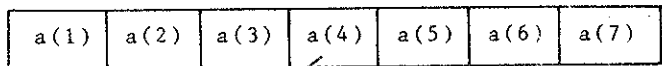


Figura 21 - Quarto elemento da matriz  $a()$  que tem sete elementos

De uma forma geral para variáveis numéricas ou STRINGS escreve-se:

```
DIM nome (dimensão)
```

com a qual permite-se reservar na memória do TK 90X área para matriz numérica ou STRING, unidimensional ou multidimensional.

Uma variável STRING deve ser dimensionada em número (x) e em comprimento (y).

Desta forma quando se escreve

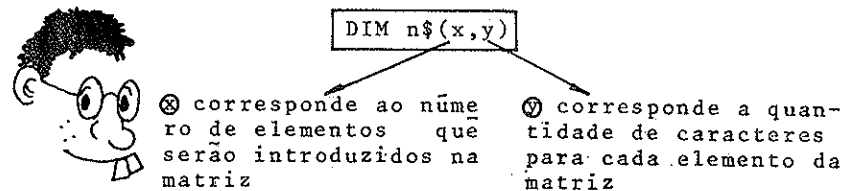
```
DIM n$(4,15)
```

está se dimensionando uma matriz para 4 STRINGS de comprimento máximo igual a 15 caracteres.

Poderiam ser elementos desta matriz, os seguintes

- $n$(1) \rightarrow$  MIRSHAWKA (9 caracteres específicos e os outros são espaços em branco)
- $n$(2) \rightarrow$  SERGIO
- $n$(3) \rightarrow$  VICTOR
- $n$(4) \rightarrow$  ALEXANDRE

De uma forma geral quando se dimensiona uma matriz STRING escreve-se



Quando para um dado elemento o número de caracteres é superior a y eles são ignorados e os elementos da matriz  $n$( )$  tem as suas posições preenchidas da esquerda para a direita e caso haja folga, as restantes posições são automaticamente preenchidas com espaços em branco.

As variáveis subscriptas facilitam e dão

grande versatilidade à programação em BASIC para o seu TK 90X.

Uma das mais importantes aplicações das variáveis subscritas é na armazenagem de lista de números e/ou STRINGS através das instruções READ-DATA ou INPUT.

A variável de controle pode também ser utilizada como variável para o subscrito em uma variável indexada, fazendo com que cresça em uma unidade, cada vez que o "loop" for lido.

Veja o programa 13 do Capítulo 23.

Matrizes numéricas bidimensionais, tridimensionais, etc.

O seu TK 90X além de aceitar "linhas de variáveis" (veja a Figura 21) pode também aceitar variáveis organizadas em tabelas feitas de linhas e colunas.

Assim por exemplo

```
DIM a(10,10)
```

define uma coleção de variáveis organizadas em 10 linhas e 10 colunas.

É comum referir-se a um elemento desta matriz ou tabela como a(i,j) onde o primeiro índice (i) indica a linha e o segundo índice (j) indica a coluna.

É evidente que com DIM a(10,10) reservamos na memória do TK 90X espaço para  $10 \times 10 = 100$  variáveis numéricas.

A idéia de uma matriz bidimensional pode ser expandida para matrizes a três, quatro, ... até 255 dimensões.

No momento não há nenhuma vantagem em "divagar" sobre o que representariam variáveis com tantos índices.

Aceite entretanto, crédulo(a) leitor(leitora) que elas tem muito uso...

Só como ilustração, nesta linha de raciocínio se você escrever por exemplo

```
DIM a(10,15,8)
```

estará reservado na memória do TK 90X espaço para  $10 \times 15 \times 8 = 1200$  variáveis numéricas sendo que um elemento típico poderia ser a(3,7,6).

Aí já surge um indício do porque matrizes com muitas dimensões não são muito usadas - você pode acabar com a memória do seu TK 90X rapidamente.

Veja nos programas 14 e 15 do Capítulo 23 a aplicação da instrução DIM e das matrizes.

Muy amigo(a) acho que quem chegou aqui vai entender tudo o que resta ou seja o Capítulo 23.

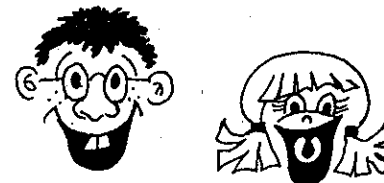
Porém não esqueça que existem outros livros de nossa autoria e dos meus filhos Sergio e Victor nos quais aparecem programas bastante originais e que lhe darão a condição de completar o seu conhecimento e o pleno domínio do TK 90X.

Por enquanto já "existem"

"Jogos e Desenhos no TK 90X"-vol.1

"Gráficos no TK 90X"

e vem muito mais aí pela frente...



## Capítulo 23

### Programas adicionais

- 1) Elabore um programa que entre com sete números através do teclado e apresente na tela a sua soma.

Solução:

```
10 INPUT N1
20 INPUT N2
30 INPUT N3
40 INPUT N4
50 INPUT N5
60 INPUT N6
70 INPUT N7
```

Aquí também poderia-se ter usado para nome das variáveis numéricas as letras A,B,C,D,E,F,G no lugar de N1,N2,N3,N4,N5,N6,N7

```
80 LET S=N1+N2+N3+N4+N5+N6+N7
90 PRINT A soma é igual a:□";S
```

Ordena-se ao TK 90X atribuir à variável S a soma das variáveis N1,N2,...,N7



Aquí se manda imprimir a mensagem e o valor da soma

- 2) Elabore um programa que some os dois primeiros números que entram, divide os mesmos por dois, a seguir soma o resultado com um terceiro número multiplicado por dois e divide esta última soma por 3 apresentando na tela o resultado final.

Solução:

Na realidade queremos obter o valor de M obtido da fórmula



$$M = \frac{\frac{A+B}{2} + 2 \times C}{3}$$

```

10 INPUT "Entre com o primeiro número";A
20 INPUT "Entre com o segundo número";B
30 INPUT "Entre com o terceiro número";C
40 LET M = ((A+B)/2 + 2*C)/3
50 PRINT "O resultado final é: ";M

```

3) Elaborar um programa que lhe pergunte neste ordem

- a) o seu nome;
- b) o seu sobrenome;
- c) o seu telefone;
- d) o seu endereço;
- e) a sua profissão;
- f) a sua idade.

Deve sair impressa a sua "ficha" completa.

#### Solução:

```

10 INPUT "Entre com o seu nome"; n$
20 INPUT "Entre com o seu sobrenome"; s$
30 INPUT "Entre com o seu telefone"; t$
40 INPUT "Entre com o seu endereço"; e$
50 INPUT "Entre com a sua profissão"; p$
60 INPUT "Entre com a sua idade"; I
70 PRINT "O meu nome é ";n$; " "; s$
80 PRINT "Meu telefone é: "; t$
90 PRINT "Moro na "; e$
100 PRINT "A minha profissão é: "; p$
110 PRINT "Atualmente tenho ";I; "anos"

```

aqui sai impresso o nome e o sobrenome

O.I. - Na linha 30 tem-se uma variável STRING "t\$" apesar de que a entrada seja um número.

O seu TK 90X guardará o número telefônico como uma cadeia de caracteres incluindo os espaços em branco.

4) Aí vai um programa que permite imprimir o dia da semana entrando-se com um número de 1 a 7.

#### Solução:

```

10 INPUT "Entre com dia da semana como um número de 1 a 7";
d
20 CLS: PRINT "Dia";d; "é";
30 GOTO 10*d+40
40 GOTO 10
50 PRINT "segunda-feira": GOTO 10
60 PRINT "terça-feira": GOTO 10
70 PRINT "quarta-feira": GOTO 10
80 PRINT "quinta-feira": GOTO 10
90 PRINT "sexta-feira": GOTO 10
100 PRINT "sábado": GOTO 10
110 PRINT "domingo": GOTO 10

```



Como se vê neste programa o número após o GOTO é substituído por uma expressão numérica.

Isto é muito útil quando se quer introduzir um elemento de escolha ou opção para o salto.

Assim devido a linha 30 o desvio será para a linha 50 se você entrar com o número "1", para a linha 60 se você entrar com o número "2"... para a linha 110 se você entrar com o número "7".

Neste programa se você entrar com o "0" então devido a instrução da linha 30 você irá para a linha 40 que o devolverá para a linha 10 e aí começa tudo de novo.

Caso o seu engano seja maior e você entre com um número maior que 7 então devido a instrução da linha 30 de-se desviar para um número de linha maior do que os que existem no programa.

Felizmente, o TK 90X tem um BASIC bastante tolerante para tais tipos de erros e se você mandou que ele

saltasse para um número de linha que não existe então ele pula para a linha mais próxima com numeração maior ou se não existe nenhuma então o programa simplesmente pára.

O.I. - Usaremos muito os dois pontos (:) que apareceram pela primeira vez, desde que você esteja respeitando a sequência indicada no livro...

O uso dos dois pontos (:) nos permite ter o que se chama linha com instruções múltiplas.

Neste caso as linhas 50, 60, 70, 80, 90, 100 e 110 todas tem duas instruções cada uma.

- 5) Elaborar um programa que permita exibir uma tabela de conversão de números binários de 8 bits nos seus equivalentes decimais.

Caso você tenha a possibilidade de conectar o seu TK 90X com uma impressora (uma boa opção seria ter uma GRAFIX...) isto lhe permitirá ter uma tabela que lhe pode ser muito útil no futuro, entretanto não esqueça de mudar o PRINT para LPRINT.

Deve-se usar obrigatoriamente neste programa um super aninhamento ou seja 8 "loops" FOR-TO/NEXT.

#### Solução:

Aí está o programa:

```
10 FOR a = 0 TO 1
20 FOR b = 0 TO 1
30 FOR c = 0 TO 1
40 FOR d = 0 TO 1
50 FOR e = 0 TO 1
60 FOR f = 0 TO 1
70 FOR g = 0 TO 1
```



```
80 FOR h = 0 TO 1
90 PRINT a*128 + b*64 + c*32 + d*16 + e*8 + f*4 + g*2 + h;
    TAB 5; "="; a;b;c;d;e;f;g;h
100 NEXT h
110 NEXT g
120 NEXT f
130 NEXT e
140 NEXT d
150 NEXT c
160 NEXT b
170 NEXT a
```



Execute e veja se sai algo do seguinte tipo:

```

=====
00000000  = 0
00000001  = 1
00000010  = 2
00000011  = 3
00000100  = 4
00000101  = 5
00000110  = 6
00000111  = 7
00001000  = 8
00001001  = 9
00001010  = 10
00001011  = 11
00001100  = 12
00001101  = 13
00001110  = 14
00001111  = 15
00010000  = 16
00010001  = 17
00010010  = 18
00010011  = 19
00010100  = 20
00010101  = 21
00010110  = 22
00010111  = 23
00011000  = 24
00011001  = 25
00011010  = 26
00011011  = 27
00011100  = 28
00011101  = 29
00011110  = 30
00011111  = 31
=====

```

- 6) Usando três "loops" vou elaborar um programa que permite ver uma linda sequência de "padrões" coloridos que vão aumentando com o passar do tempo em complexidade e que lhe trará dedicado(a) leitor(leitora) uma distração durante uns 10 minutos.

```
10 FOR c = 50 TO 1 STEP -1
20 FOR a = 0 TO 11
30 FOR b = a TO 19
```



```

40 LET d = INT(a*b/c+(a+b)/c+c) : PAPER d-9*INT(d/9)
50 PRINT AT a,b; "□"; AT a,31-b; "□"; AT 21-a,b; "□",
  AT 21-a, 31-b; "□"
60 PRINT AT b,a; "□"; AT b,31-a; "□"; AT 21-b,a; "□";
  AT 21-b,31-a; "□"
70 NEXT b
80 NEXT a
90 NEXT c

```

Execute e sinta-se um(a) verdadeiro(a) artista.

Pode até exclamar exótico, marcante, criativo!!!

**O.I.-** 1) PAPER código de cor (linha 40) define a cor (de 0 a 7) do fundo da tela.

2) INT é uma função que retorna a parte inteira de um número fracionário.

3) PRINT AT y,x desloca a posição de impressão para uma genérica coluna x e uma genérica linha y sendo  $0 \leq x \leq 31$  e  $0 \leq y \leq 2$ .

7) Embora este programa não seja muito longo ele lhe dará como resultado um jogo bem excitante.

O quadrado a esquerda (Figura 22) tem uma letra dentro do mesmo.

Quando ele se desloca para o outro lado da tela você precisa tentar parar o mesmo na região indicada pela mesma letra.

No início do jogo o TK 90X lhe pergunta em que nível de dificuldade (quanto mais rápido tanto mais difícil...) você quer participar do jogo e a sua escolha pode ser:

- |   |                                  |
|---|----------------------------------|
| { | 1 → bem fácil (lento)            |
|   | 2 → difícil (rápido)             |
|   | 3 → muito difícil (muito rápido) |

Colocou-se um sistema de contagem e no final aparece o número total de acertos.

```

10 REM *** Pare-o no lugar certo ***
15 BORDER 5 : PAPER 7 : INK 2
20 LET s$ = "abcd□" : LET t = 0 : LET a = 0
30 INPUT "Com nível de dificuldade (1-3) : "; di : LET d =
  10 - 3*di
40 INPUT "Quantas tentativas você fará?"; n
50 FOR s = 1 TO 5
60 FOR l = 0 TO 15 : PRINT AT l,14+3*s; PAPER 0; "□";
  : NEXT l
70 PRINT AT 0,15+3*s; s$(s) : NEXT s
80 LET r = INT (RND*4+1) : LET l$ = s$(r)
90 LET t = t+1 : LET l = 2 + 2*INT(RND*7) : LET c = 0
100 PRINT AT l,c; "□"; l$ : FOR J = 1 TO d : NEXT J
110 LET c = c+1 : IF c < 31 AND INKEY$="" THEN GOTO 100
120 IF INT((c-15)/3) <> r OR c = 17+3*r THEN GOTO 140
130 SOUND .1,11 : LET a=a+1 : GOTO 150
140 SOUND .1,0 : LET r$ = INKEY$
150 PRINT AT 20,0; "a contagem é:□"; a; "□acertos em□";
  t; "□tentativas"
160 IF t < n THEN GOTO 80

```

Execute e boa sorte.

Tenha muito bom senso, lógica e velocidade no dedo.

Note que:

- 1) bom senso é a lógica natural ou seja aptidão inata da inteligência para descobrir a verdade.
- 2) lógica é a ciência que estuda as leis do pensamento e a arte de aplicá-las corretamente na investigação e na demonstração da verdade.





3) dedo você tem, não é?

O bom senso e a lógica você irá aplicar para que a letra pare na zona certa como está indicado na Figura 22.

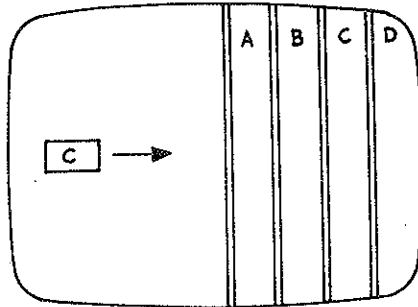


Figura 22

8) Vamos mostrar neste programa a potencialidade do seu TK 90X para produção de figuras.

O.I. Consulte o nosso livro "Gráficos no TK 90X" para maiores detalhes sobre este tema.

a) Obtém-se uma série de circunferências que no conjunto dão a impressão de se ter um cone

```
10 BORDER RND *6+1: PAPER RND *6+1: INK 0 : CLS
20 LET y = 1 : LET r = 1
30 FOR x = 1 TO 180 STEP 5
40 CIRCLE x,y,r
50 LET y = y+2 : LET r = r+2
60 NEXT x
```



Execute e veja se sai algo deste tipo

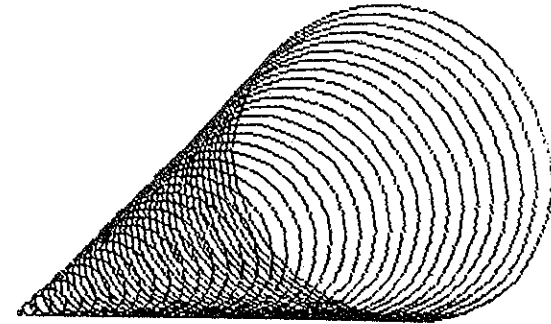


Figura 23

b) Neste programa desenham-se linhas retas de tal forma que elas acabam formando uma curva.

É você caro(a) usuário(a) que deve indicar a cor da tinta e o espaçamento entre as retas.

```
10 INPUT "Entre com a cor da tinta?"; c
20 BORDER 7 : PAPER 7 : INK c : CLS
30 INPUT "Entre com um número de 1 até 25 "; e
40 LET a = 0
50 PLOT 0,175 : DRAW 0, -175
60 PLOT 0,0 : DRAW 175,0
70 FOR j = 175 TO 0 STEP -e
80 PLOT 0,j : DRAW a,-j
90 LET a = a+e
100 NEXT j
```

Execute e veja se sai algo deste tipo:

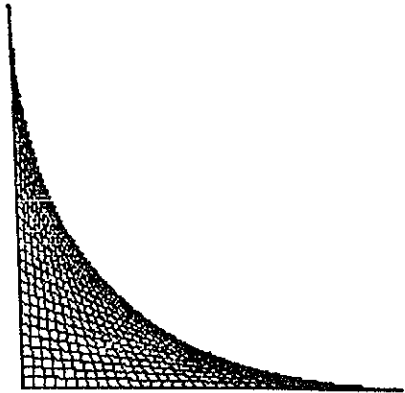


Figura 24

- 9) Elaborarei agora uma série de pequenos programas para des-  
tacar a influência e o uso conjunto do PLOT e do DRAW per-  
mitindo obter figuras de contorno fechado como por exem-  
plo um quadrado ou uma simulação dos raios solares.

```
a) 10 REM *** O nascer do Sol ***
    20 BORDER 5 : PAPER 1 : INK 6
    30 FOR j = 1 TO 100
    40 PLOT 127,87 : DRAW 100*RND, 70*RND
    50 PLOT 127,87 : DRAW -100*RND, 70*RND
    60 PLOT 127,87 : DRAW -100*RND, -70*RND
    70 PLOT 127,87 : DRAW 100*RND, -70*RND
    80 NEXT j
    90 PRINT INK 2; FLASH 1; AT 21,3; "Que lindo Sol!!!"
    100 FOR z = 1 TO 499 : NEXT z : CLS : GOTO 30
```



Execute e veja se sai algo parecido com o que está in-  
dicado na Figura 25.

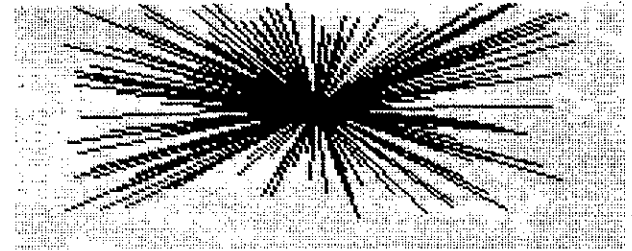


Figura 25

```
b) 10 REM *** Vendo um Sol duplo ***
    20 BORDER 1 : PAPER 6 : INK 2
    30 FOR k=1 TO 3 STEP 2
    40 FOR j=1 TO 80
    50 PLOT 60*k, 87 : DRAW 55*RND, 75*RND
    60 PLOT 60*k, 87 : DRAW -55*RND, 75*RND
    70 PLOT 60*k, 87 : DRAW -55*RND, -75*RND
    80 PLOT 60*k, 87 : DRAW 55*RND, -75*RND
    90 NEXT j
    100 NEXT k
    110 PAUSE 100
    120 PRINT INK 3; FLASH 1; AT 21,4; "Mais do que um
        Sol??????"
    130 PAUSE 100
    140 CLS: GOTO 30
```

Execute e admire dois "astros-rei" ao mesmo tempo.

```
c) 10 REM *** desenhando um quadrado ***
    15 LET c = 0
    20 BORDER 0 : PAPER 1 : INK c
```

```

30 PRINT INK 2; FLASH 1; AT 21,5; "Um quadrado
   isolado"
40 PLOT 50,50 : DRAW 50,0 : DRAW 0,50 : DRAW -50,0 :
   DRAW 0,-50
50 PRINT INK 3; FLASH 1; AT 3,7; "Você quer ver um
   quadrado de outra cor? (S/N)□": INPUT r$:
   IF r$ = "n" OR r$ = "N" THEN GOTO 90
60 PRINT INK 3; FLASH 1; AT 5,7; "Entre com o código
   de 2 a 7";
70 INPUT c
80 GOTO 20
90 STOP

```

```

d) 10 REM *** coleção de quadrados ***
20 BORDER 0: PAPER 7: INK 1
30 FOR j = 10 TO 80 STEP 10
40 PLOT j,j: DRAW j,0: DRAW 0,j: DRAW -j,0:
   DRAW 0,-j
50 NEXT j
60 PRINT INK 2; FLASH 1, AT 0,0; "Que linda coleção
   de quadrados"
70 FOR p = 1 TO 399 : NEXT p : CLS
80 GOTO 30

```



Gostou destes mini-programas?

Sem dúvida que sim, pois o mistério das instruções  
PLOT e DRAW se desfaz diante de tanta simplicidade.

10) Agora realmente você admirará o seu TK 90X que em um mi-  
núsculo programa lhe mostrará um interessantíssimo dese-  
nho.

```

5 BORDER 0 : PAPER 7 : INK RND*6 + 1
10 FOR j = 0 TO 1.7*PI STEP PI/3
20 LET x = 127: LET y = 88 : LET r = 100

```

```

30 FOR k = j TO j + 1 STEP .1
40 PLOT x,y : DRAW r*COS k, r*SIN k
50 DRAW r*SIN(PI/6-k) - r*COS k, r*COS(PI/6-k)-r*SIN k
60 DRAW - r*SIN(PI/6-k), -r*COS(PI/6-k)
70 LET x = x+.1*r*COS k: LET y=y+.1*r*SIN k : LET r=.85*r
80 NEXT k
90 NEXT j

```

Execute e veja se sai algo como isto:

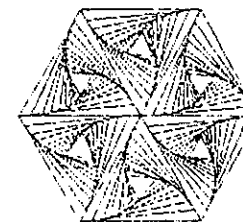


Figura 26

11) Com este programa iremos obter os caracteres escritos em  
tamanho bem grande e depois com pequenas modificações fa-  
remos aparecer um texto com caracteres bem grandes.

```

1 REM * escrevendo em tamanho maior***
5 PAPER 6 : INK 1 : BORDER 1
10 INPUT "Escolha o seu caracter□"; c$
20 IF LEN c$ > 1 THEN GOTO 10
25 LET ei = PEEK 23606 + 256 * PEEK 23607 + 256
35 LET ec = ei + 8 * (CODE c$ - 32)
45 FOR j = 0 TO 7
50 LET f = PEEK (ec + j)
60 FOR i = 0 TO 7
70 LET b = f - 2*INT(f/2) : LET f = INT(f/2)
75 IF b = 0 THEN GOTO 90
80 PRINT AT 21, (20-2*i); "■"
90 NEXT i
100 PRINT
110 NEXT j
100 PRINT ' '

```

Veja o que significa a instrução CODE no programa 27 do capítulo 21

CAPS SHIFT + GRAPHICS +8 segurando a CAPS SHIFT

Para passar da impressão de um único caractere a uma cadeia de caracteres basta fazer as seguintes modificações:

- 1) → 10 INPUT "Entre com o texto□"; c\$
- 2) → Eliminar a linha 20
- 3) → 30 FOR k = 1 TO LEN c\$
- 4) → 40 LET ec = ei + 8 \* (CODE (c\$(k TO k)) - 32) o que é LEN
- 5) → 120 NEXT k

**O.I.** Não esqueça que quando aparece "scroll" basta apertar qualquer coisa que a tela sobe.

- 12) Neste programa, usando READ, DATA, RESTORE, PLOT, DRAW, INVERSE, OVER etc. desenharemos uma cápsula espacial (desenho aproximado) apresentando-a 32 vezes na tela conforme se mostra na Figura 27.

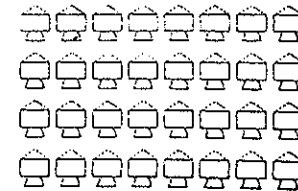
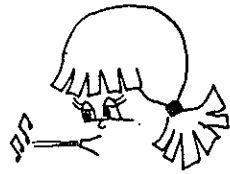


Figura 27

Aí vai o programa:

```

5 BORDER 1 : PAPER 5 : INK 0
10 REM***Desenhando cápsulas***
20 FOR v = 15 TO 135 STEP 40
30 FOR h = 25 TO 235 STEP 30
40 PLOT INVERSE 1; OVER 1; h,v
50 RESTORE
60 READ m,x,y
70 IF m = 0 THEN DRAW INVERSE 1; OVER 1; x,y
80 IF m = 1 THEN DRAW x,y
90 IF m <> 9 THEN GOTO 60
100 NEXT h
110 NEXT v
200 DATA 0,-12,7,1,24,0,1,0,-14,1,-24,0,1,0,14
210 DATA 1,12,8,1,12,-8
220 DATA 0,-7,-14,1,3,-8,1,-16,0,1,3,8,9,0,0

```

Com esta instrução desloca-se a posição de desenho para o centro do próximo desenho sem que com isto se faça alguma marca na tela



13) Aí vão 3 aplicações nas quais se utiliza a variável subscrita.

a) Neste programa você pode entrar com 15 números e os mesmos serão arranjados em ordem crescente e serão exibidos juntamente com o seu logaritmo natural.

Caso o seu número seja negativo para o logaritmo surge a mensagem "Não existe?"

```

10 REM***Ordenação numérica***
20 DIM n(15)
30 BORDER 0; PAPER 6; INK 1
40 FOR j = 1 TO 15
50 INPUT n(j)
60 IF j=1 THEN GOTO 130
70 FOR k = j TO 2 STEP -1
80 IF n(k) >= n(k-1) THEN GOTO 120
90 LET aux = n(k)
100 LET n(k) = n(k-1)
110 LET n(k-1) = aux
120 NEXT k
130 NEXT j
140 PRINT "Valor", "Logaritmo(e)"
150 FOR i = 1 TO 15
160 IF n(i) < 0 THEN SOUND 3,20: GOTO 190
170 PRINT n(i), LNn(i)
180 GOTO 200
190 PRINT n(i), "Não existe?"
200 NEXT i

```



b) Agora você verá um uso intenso do READ junto com variáveis indexadas do tipo STRING.

```

10 REM *** escolhendo um mês ***
20 DIM m$(12,3)
30 DATA "jan","fev","mar","abr","mai","jun","jul","ago",
"set","out","nov","dez"
40 FOR j = 1 TO 12
50 READ m$(j)
60 NEXT j
70 LET r = INT(RND*12) + 1
80 PRINT FLASH 1; AT 11,10; m$(r)
90 PAUSE 0 : SOUND .03,20
100 LET e$ = INKEY$
110 IF e$ = CHR$ 13 THEN GOTO 90
120 GOTO 70

```

O.I. → Experimente mudar a linha 20 para  
 20 DIM m\$(12,2)  
 e veja o que ocorre.  
 Houve algum corte?



c) Neste programa faremos do seu TK 90X um "inteligente" criador de textos (as vezes desconexos).

```

10 REM*** Criando frases ***
20 DIM f$(6,32) → reserva-se espaço para 6 variáveis STRING cada uma podendo ter até 32 caracteres
30 DIM p$(35,18) → reserva-se espaço para 35 variáveis cada uma podendo ter até 18 caracteres
40 FOR i=1 TO 35
50 READ p$(i)
60 NEXT i
70 DATA "Maceio","Cabo Frio", "Ponta Grossa", "Ibiuna",
"Piedade"
80 DATA "machucou", "feriu", "contundiu", "quebrou",
"torceu"
90 DATA "ombro", "joelho", "braço", "dedo", "pé"

```

```

100 DATA "no jogo", "com a porta", "na escola", "na
discussão", "no rebuliço"
110 DATA "sofrendo", "agoniando", "passando mal", "se
machucando", "se estorçalhando"
120 DATA "tanto", "desta forma", "daquela maneira",
"deste jeito", "deste modo"
130 DATA "por obra do acaso", "sem merecer", "por
puro azar.", "dolorida.", "tão grave."
140 FOR j = 1 TO 6
150 READ f$(j)
160 NEXT j
170 DATA "Era um(a) jovem que nasceu em", "e que",
"o seu"
180 DATA "Um (a) rapaz (moça) que vivia em", "em certa
época", "o seu"
190 LET a = 0
200 LET b = 0
210 LET q = INT(RND*2)+1: LET a=3*(q-1)
215 LET a = a+1
220 IF a = 3*q+1 THEN GOTO 290
230 PRINT f$(a)
240 LET b = b+1
250 LET r = INT(RND*5+1): LET k = 5*(b-1)+r
260 PRINT p$(k)
270 IF b >= 3 AND b < 7 THEN GOTO 240
280 GOTO 215
290 INPUT "Mais uma vez (S/N)?"; a$
300 IF a$ = "S" THEN RUN

```

#### Alguns comentários rápidos sobre este programa

Linhas 190 e 200 → servem de inicialização

Linhas 210 e 260 → servem para que se imprima o conteúdo guardado nas variáveis STRING f\$ e p\$ respectivamente

Linha 210 → serve para que se possa escolher um dado entre aqueles armazenados na matriz f\$( )

Linha 250 → serve para que se possa escolher aleatoriamente uma palavra ou frase entre aquelas armazenadas na matriz p\$( ).

Linha 270 → serve para que se possa imprimir o conteúdo de duas variáveis da matriz p\$( ) antes de se tomar um novo elemento de f\$( ).

Linha 280 → serve para que se imprima o conteúdo de duas variáveis da matriz f\$ antes de se tomar um novo elemento de p\$( ).

- 14) Neste programa tem-se um pequeno jogo instrutivo no qual o seu TK 90X lhe exhibe uma palavra em português e pede que você entre com a correspondente em inglês e vice versa.

É claro que nos limitaremos aqui a 30 nomes porém você pode ampliar em muito esta lista...

Perdoe-me caso você indique algum sinônimo e ele não seja aceito no programa.

```

10 REM**Português X Inglês**
20 RAND
30 LET c = 0
40 LET p = 0
50 DIM p$(30,10) : DIM i$(30,10)
60 FOR j=1 TO 30
70 READ p$(j)
80 READ i$(j)
90 NEXT j
100 INPUT "Como te chamas?"; n$
110 IF LEN n$ > 9 THEN LET n$ = n$(TO 9)
120 PAUSE 250
130 CLS

```



```

140 PRINT PAPER 6; INK 1; AT 0,4; "PONTOS:□"; c
150 PRINT PAPER 6; INK 1; AT 0,16; "PERGUNTAS:□"; p
160 LET r = INT (RND*30)+1
170 IF RND >.5 THEN GOTO 330
180 PRINT PAPER 4; INK 1; AT 6,4; "Qual é a palavra
    inglesa";
190 PRINT PAPER 4; INK 1; AT 7,4; "para□"; p$(r);
    "□ ?□□□□□□□"
200 INPUT r$
210 LET p = p+1
220 IF LEN r$ < 10 THEN LET r$ = r$ + "□" : GOTO 220
230 IF r$ = i$(r) THEN GOTO 280
240 SOUND .125,-7 : SOUND .5,-9 : SOUND .1,-11
250 PRINT PAPER 3; INK 1; AT 12,4; "Não□"; n$; "□ a
    palavra é"
260 PRINT PAPER 3; INK 1; AT 13,4; i$(r); "□□□□□□□□□□"
270 GOTO 120
280 SOUND .125,9 : SOUND .25,8 : SOUND .5,7 : SOUND 1,6
290 PRINT PAPER 3; INK 1; AT 10,4; "Parabéns está correto!"
300 LET c = c+1
310 IF c = 30 THEN GOTO 430
320 GOTO 120
330 PRINT PAPER 4; INK 1; AT 6,4; "Qual é a palavra
    portuguesa"
340 PRINT PAPER 4; INK 1; AT 7,4; "para□"; i$(r); "□ ?
    □□□□□□□□□□ "
350 INPUT r$
360 LET p = p+1
370 IF LEN r$ < 10 THEN LET r$ = r$ + "□" : GOTO 370
380 IF r$ = p$(r) THEN GOTO 230
390 SOUND .25,0 : SOUND .25,2 : SOUND .25,4 : SOUND .25,5
400 PRINT PAPER 3; INK 1; AT 12,4; "Não□"; n$; "□ a pa-
    lavra é □"

```

```

410 PRINT PAPER 3; INK 1; AT 13,4; p$(r)
420 GOTO 120
430 PAPER 0 : BORDER 0 : CLS
440 PRINT PAPER 5 ; INK 1; AT 2,2; "Até que enfim!!!"
450 PRINT PAPER 5 : INK 1 : AT 3,2: "Você conseguiu
    acertar 30"
460 PRINT PAPER 5 : INK 1 : AT 4,12; "perguntas"
470 FOR r = 1 TO 36 STEP 3
480 CIRCLE INK 6; 125, 70, r
490 NEXT r ;
500 PAUSE 0 : PAPER 7 : BORDER 7
510 CLS
520 STOP
530 DATA "macaco", "ape"
540 DATA "braço", "arm"
550 DATA "mau", "bad"
560 DATA "belo", "beautiful"
570 DATA "preto", "black"
580 DATA "carvão", "coal"
590 DATA "cruz", "cross"
600 DATA "fundo", "deep"
610 DATA "diamante", "diamond"
620 DATA "terra", "land"
630 DATA "envelope", "envelope"
640 DATA "campo", "field"
650 DATA "alimento", "food"
660 DATA "genio", "genius"
670 DATA "chapeu", "hat"
680 DATA "interior", "indoor"
690 DATA "beijo", "kiss"
700 DATA "trabalho", "labor"
710 DATA "máquina", "machine"
720 DATA "norte", "north"

```

```

730 DATA "ostra", "oyster"
740 DATA "pico", "peak"
750 DATA "restaurar", "restore"
760 DATA "sardinha", "sardine"
770 DATA "irmandade", "sorority"
780 DATA "tarefa", "task"
790 DATA "trinta", "thirty"
800 DATA "vencedor", "victor"
810 DATA "voz", "voice"
820 DATA "valsa", "waltz"

```

15) Este programa é um pequeno quebra-cabeça ou "raxa kuka" e vou-lhe indicar inicialmente as normas a seguir.

Para começar, talvez necessite de lápis e papel, porém se você proceder assim seria como se você estivesse trapacéando.

O objetivo é achar a área de um trapézio ou seja a área de um triângulo somada a área de um retângulo.

O programa lhe perguntará "Um ou dois jogadores?".

Tecler  ou  e aí pressione a .

Em seguida lhe perguntará o nome ou os nomes dos jogadores e depois de ter introduzido as informações requeridas tecler de novo .

Aí o TK 90X lhe mostrará um retângulo verde com um triângulo azul em cima.

Você será informado(a) sobre o comprimento de cada um dos lados e lhe será pedido para calcular a área total.

Caso você não consiga dar a resposta correta inicialmente o TK 90X lhe perguntará só sobre a área do retângulo.

Se desta vez a sua resposta for correta, então mostrará o triângulo de novo perguntando-lhe a respeito da área só do triângulo.

Caso alguma das suas respostas (triângulo ou retângulo) esteja errada, o TK 90X lhe mostrará a área total correta.

Entretanto se a sua primeira resposta a (total) estiver correta, está tudo terminado "gatão" ("gatinha") e como prêmio vais ganhar 2 pontos, porém se acertas em duas vezes só terá 1 ponto.

Aí vai uma "dica": a área total pode ser obtida pela fórmula

$$B \times C + \frac{1}{2} B \times (A - C)$$

Bem aí vai o programa propriamente dito:

```

10 REM ** Áreas **
15 BORDER 3: PAPER 6
20 LET d = 40 : LET e = 90
30 DIM s(2)
40 DIM p$(2,9)
50 RAND : LET s(1) = 0 : LET s(2) = 0
60 INPUT "1 ou 2 jogadores?"; n
70 IF n < 1 OR n > 2 THEN GOTO 60
80 PRINT AT 10,3; "Quais são os nomes dos jogadores?"
90 FOR j=1 TO n
100 PRINT AT 13,4; "Jogador";j
110 INPUT p$(j)
120 NEXT j
130 CLS
140 FOR z = 1 TO n
150 LET a = INT(RND*7) + 4
160 LET b = INT (RND*8) + 3
170 LET c = INT (RND*5) + 3
180 IF c >= a THEN GOTO 150
190 LET sa = a*8
200 LET sb = b*8
210 LET sc = c*8

```

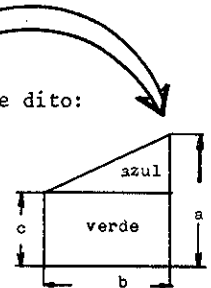


Figura 28





```

220 FOR j = d TO d - 1 + sc
230 PLOT e, j
240 DRAW INK 4; sb, 0
250 NEXT j
260 PLOT e, d+sc
270 DRAW INK 1; sb, sa-sc
280 DRAW INK 1; 0, - (sa-sc)
290 FOR j = d + sc TO d-1 + sa
300 FOR k = e - 1 + sb TO e STEP-1
310 IF POINT (k, j) = 1 THEN LET k = 60: GOTO 340

330 PLOT INK 1; k, j
340 NEXT k
350 NEXT j
360 PRINT AT 16-INT a/2, 13+b;a
370 PRINT AT 18, INT 11 + b/2; b
380 PRINT AT INT 16 - c/2, 9; c
390 PRINT AT 20, 0; p$(z); ", qual é a área total?"
400 INPUT resposta
410 IF resposta = c*b + b/2 * (a-c) THEN GOTO 680
420 SOUND .8, -8 : SOUND .1, -10 : SOUND .125, -12
430 PRINT AT 20, 0; p$(z); ", qual é a área do retângulo?"
"□□□□□□□□□□□□□□□□□□"
440 INPUT ret
450 IF ret = c*b THEN GOTO 500
460 SOUND .8, -8 : SOUND .1, -10 : SOUND .125, -12
470 PRINT AT 20, 0, "A área do retângulo é "; c*b;
"□□□□□□□□□□□□□□□□□□"
480 PAUSE 220
490 GOTO 640
500 PRINT AT 20, 0; p$(z); ", qual é a área do triângulo?"
"□□□□□□□□□□□□□□□□□□"
510 INPUT triang
520 IF triang = b/2 * (a-c) THEN GOTO 570

```

Esta função informa se uma determinada posição da tela (coluna x, de 0 a 255 e linha y de 0 a 175) foi desenhada ou ativada voltando 1 se está na cor de INK e 0 se está na cor do PAPER

```

530 SOUND .8, -8 : SOUND .1, -10 : SOUND .125, -12
540 PRINT AT 20, 0; "A área do triângulo é "; b/2*(a-c);
"□□□□□□□□□□□□□□□□□□"
550 PAUSE 220
560 GOTO 640
570 PRINT AT 20, 0; p$(z); ", qual é a área total?"
"□□□□□□□□□□□□□□□□□□"
580 INPUT resposta
590 IF resposta = b*c + b/2 * (a-c) THEN GOTO 620
600 SOUND .8, -8 : SOUND .1, -10 : SOUND .125, -12
610 GOTO 640
620 LET s(z) = s(z) + 1 : SOUND .125, -5 : SOUND .125, -7 :
SOUND .125, -9
630 GOTO 660
640 PRINT AT 20, 0; "A área total é □"; b*c + b/2 * (a-c);
"□□□□□□□□□□□□□□□□□□"
650 PAUSE 220
660 CLS
670 GOTO 700
680 SOUND .25, 0 : SOUND .25, -1 : SOUND .25, -3 :
SOUND .25, -5
690 LET s(z) = s(z) + 2
700 PRINT AT 1, 4; p$(1); "□:□"; s(1); "□□□"; p$(2);
"□:□"; s(2)
710 NEXT z
720 GOTO 140

```

# ÍNDICE REMISSIVO

alta resolução, 137  
AND, 125  
ASCII, 173  
aspas, 9  
AT (veja PRINT AT, 94)

BASIC, 1  
BIN, 186  
bit, 194  
byte, 194  
BORDER, 77  
BREAK, 69

C (cursor), 8  
CAPS LOCK, 10  
CAPS SHIFT, 10  
caracteres gráficos, 129  
cassete, 55  
CHR\$, 165  
CIRCLE, 146  
CLEAR, 177  
CLS, 71  
CODE, 175  
cursor, 8

DATA, 151  
DELETE, 11  
DIM, 200  
dois pontos (":"), 123  
DRAW, 141

E (cursor), 8  
EDIT, 38  
endereço, 24  
ENTER, 5  
espaço de trabalho, 11,82

FLASH, 135  
flecha, 39  
FOR-TO/NEXT, 109  
função corte, 106

G (cursor), 8  
GOSUB, 88  
GOTO, 64  
GRAPHICS, 136

IF/THEN, 68  
INK, 77  
INKEY\$, 124  
INPUT, 41  
INT, 66  
INVERSE, 139

K (cursor), 8

L (cursor), 8  
LEN, 166, 178  
LET, 21  
LINE, 172  
linguagem de alto nível, 1

linguagem de máquina, 175  
 LIST, 49  
 LN, 71  
 LOAD, 55  
 LOAD "", 5  
 LOAD "teste", 154  
 LOOP, 117  
  
 matriz, 199  
 memória, 25  
 MIC, 56  
 microprocessador, 22  
 modo imediato, 31  
 modo programado, 31  
 multidimensional, 201  
  
 NEW, 35  
 NOT, 173  
 número aleatório, 63  
 número de linha, 31  
  
 operadores, 14  
 operadores lógicos, 173  
 operadores relacionais, 67  
 OR, 173  
 OVER, 219  
  
 PAPER, 77  
 PAUSE, 123  
 palavra chave, 2  
 PEEK, 149  
 PI, 145  
 pixel, 137  
 PLOT, 138  
 POKE, 186  
 PRINT, 6  
 PRINT AT, 94  
 PRINT "", 39  
 PRINT#, 197  
 programa, 31  
  
 RAM, 23  
 RAND, 68  
 READ, 152  
 REM, 68  
 RESTORE, 154,178  
 RETURN, 89  
 RND, 63  
 ROM, 23  
 RUN, 33  
  
 SAVE, 55  
 SAVE "teste", 58  
 SCREEN\$, 134  
 scroll, 64  
 símbolo, 14,15,67  
 SPACE, 11  
 STEP, 107  
 STRING, 27  
 subrotina, 88  
 substring, 105  
 SYMBOL SHIFT, 9  
  
 TAB, 114  
  
 unidade central de processamento,  
 22  
 unidimensional, 201  
 UDG, 179  
 USR, 177  
  
 variável numérica, 21  
 variável STRING, 28  
 variável subscrita ou indexada,  
 200  
 VERIFY, 55  
 VERIFY "teste", 59

#### OUTRAS OBRAS DO AUTOR

Caderno de elementos de computação (Linguagem FORTRAN)  
 Linguagem BASIC  
 BASIC sem segredos  
 TK-divertindo  
 TK-lembrando  
 TK-calculando  
 Conhecendo e utilizando o TK 2000  
 TK-2000 na matemática  
 Dê um APPLE à sua vida  
 Imprimindo maravilhas com a GRAFIX  
 Brincando com o TRS-Color  
 BASIC no TK 90X  
 Conexão: GRAFIX-IBM PC e compatíveis  
 Gráficos no TK 2000  
 Jogos e Desenhos no TK 2000

0267

Gráfica Editora Guteplan Ltda.  
Rua Hugo D'Antola, 46 - Lapa - São Paulo